HACKER

Season 05 / Issue 12
τεύχος 060 - Console Attraction Edition

Διαχείριση virtual machines από command line [*]

Μπλοκάρισμα διαφημίσεων για όλο το οικιακό δίκτυο

Ανάπτυξη VM και guest OS, από τη γραμμή εντολών

Υπηρεσίες PXE server, από το Raspberry Pi

Σωστά κι έξυπνα: κλωνοποίηση QEMU/KVM VMs

Απλά κι εύκολα: μεταμορφώστε το Raspberry Pi σε VPN server

Άνετα κι αστραπιαία: προετοιμασία openSUSE VM

Αποτροπή root password reset σε συστήματα Linux

[*] γιατί σαν την κονσόλα δεν έχει

HACKEЯ

#dH060 | Περιεχόμενα

Εκδότης / Διευθυντής	Χρήστος Βαρελάς
Αρχισυντάκτης	Παναγιώτης Βαρελάς
Σύμβουλοι έκδοσης	Βούλα Παυλίδου
Σύνταξη	Παναγιώτης Βαρελάς, Χρήστος Βαρελάς, Γιώργος Γιάννου, Γιάννης Κρομμύδας, Πέτρος Κυλαδίτης, Μέσος Παπαδόπουλος (TM), Χρίστος Τόμπρας, @arkoudos
Γραμματική επιμέλεια	Βούλα Παυλίδου
Καλλιτεχνική Επιμέλεια	Πέτρος Φιλιππίδης

Το ψηφιακό περιοδικό deltaHacker είναι συνδρομητικό και εκδίδεται από την Parabing Creations. Ο δικτυακός τόπος του περιοδικού είναι στο https://deltahacker.gr και το email επικοινωνίας είναι το talk2us@deltahacker.gr. Οι γενικές πληροφορίες για τις συνδρομές παρατίθενται στο https://deltahacker.gr/order κι από την ίδια σελίδα γίνονται οι παραγγελίες.

Αερολογίες κι αυταπάτες

του Χρήστου Βαρελά

Φίλες και φίλοι,

- Κάτι που συχνά με απασχολεί έχει να κάνει με το **νόημα**.
- Όχι το νόημα της ζωής, από πού ερχόμαστε, πού πάμε, γιατί υπάρχει (;) κάτι κι όχι το τίποτα, είναι το τίποτα **κάτι**; κ.λπ.
- Αναφέρομαι απλά στη **σημασία** όσων λέμε/ακούμε ή γράφουμε/διαβάζουμε.
- Ίσως φταίει η λιγοστή Μαθηματική παιδεία που είχα πάρει αμέτρητα χρόνια πριν, ίσως κάποιος περιορισμός στην αντίληψή μου, ίσως ένας μη-γραμμικός συνδυασμός των δύο: Ό,τι και να φταίει, το θέμα είναι πως αρκετές φορές δυσκολεύομαι να καταλάβω τι διαβάζω ή τι ακούω. Η πληροφορία που εκλαμβάνω, δηλαδή, είτε δε βγάζει νόημα είτε είναι ελλιπής – και πάντα για την αντίληψή μου.
- Στα Μαθηματικά ό,τι λέμε είναι σαφώς ορισμένο, τουλάχιστον εντός του πλαισίου που ορίζει ένα πλήθος Αξιωμάτων. Στην Ευκλείδεια Γεωμετρία, για παράδειγμα, αν πούμε πως δύο ευθείες είναι ασύμβατες τότε δεν χωράει **καμία** παρερμηνεία για το τι εννοούμε. Κάποιος που δεν έχει ακούσει ξανά για αυτές τις ασύμβατες ευθείες, εντελώς δικαιολογημένα δεν θα καταλάβει ή έστω δεν θα είναι σίγουρος για το τι πράγμα μιλάμε. Αλλά εδώ είναι η ομορφιά των Μαθηματικών: ασχέτως αν κάποιος γνωρίζει περί ασύμβατων ευθειών, ασχέτως αν Εκεί Έξω (TM) υπάρχουν ευθείες, ασύμβατες ή όχι, στο πλαίσιο της Ευκλείδειας Γεωμετρίας μία πρόταση σαν την "οι ευθείες Ε1 και Ε2 είναι ασύμβατες" έχει νόημα – και καθόλου δεν πειράζει αν το κατανοούμε ή όχι. (Εντάξει, αυτό το τελευταίο θα εγείρει περί τις 129 ενστάσεις από πλευράς Θείου Ακακίου, αλλά συνειδητά επιλέγω να τον κεντρίσω και μετά απλά ν' αδιαφορήσω.)
- Αλλά η Πραγματική Ζωή (TM), φίλες και φίλοι, είναι μεγαλύτερη από τα Μαθηματικά. Ποτέ δεν το ξεχνώ αυτό. Μολαταύτα συχνά μπερδεύομαι, μερικές φορές συγχύζομαι κιόλας.

/var/log/messages

- Καταλαβαίνω βεβαίως ότι μερικές φορές οι άνθρωποι κάνουν **μεγάλη** προσπάθεια ώστε να **μην** εκπέμπουν πληροφορία. Διαφορετικά: Κάνουν ό,τι μπορούν ώστε αυτό που λένε ή γράφουν να μη σημαίνει τίποτε απολύτως. Κάποιοι μάλιστα ισχυρίζονται ότι όσο πιο γενικόλογο, ασαφές, ανακριβές ή αναληθές είναι αυτό που λες, τόσο μεγαλύτερη είναι η πιθανότητα να κερδίσεις την προσοχή σημαντικού ποσοστού των ανθρώπων στους οποίους απευθύνεσαι. Επιπρόσθετα, μεγάλο ποσοστό αυτού του **ήδη** μεγάλου ποσοστού δεν θα αργήσει να δώσει αξία στα λεγόμενά σου. Όσο πιο θρασύς είσαι, λοιπόν, τόσες περισσότερες αντιδράσεις σαν τις ακόλουθες καταφέρνεις κι αποσπάς: "ν' αγιάσει το στόμα σου" (κάπως φρικιαστική προοπτική, τώρα που το καλοσκέφτομαι), "μωρέ καλά τα λέει αυτός, γι' αυτό έχουν πέσει όλοι πάνω του να τον φάνε", "είμαστε μαζί σου, προχώρα και μη φοβάσαι τίποτε" κ.ο.κ.
- Οι πολιτικοί γνωρίζουν άριστα την τέχνη του να μιλάς χωρίς να λες τίποτε απολύτως. Χάρη μάλιστα στη θαυμαστή ικανότητά τους να **μη ντρέπονται** όταν πουλάνε αέρα κοπανιστό ή/και όταν ψεύδονται ασύστολα, συχνά καταφέρνουν κι αποκομίζουν σημαντικά οφέλη.
- Παράδειγμα: Γνωστή πολιτικός πρόσφατα εκστόμισε μια πρόταση, το περιεχόμενο της οποίας σε μια κλίμακα από 1 έως 10 είναι πολύ κοντά στο -1. Η πρόταση: "Ο κύβος ερρίφθη. Ο δρόμος για τη συγκρότηση της νέας μεγάλης παράταξης είναι ανοιχτός. Όλοι όσοι ανήκουν σε αυτό το χώρο οφείλουν να ανταποκριθούν. Αυτή είναι η απαίτηση της κοινωνικής μας βάσης, οι εξελίξεις τρέχουν, δεν υπάρχουν περιθώρια για καθυστερήσεις."
- Εδώ έχουμε ένα κλασικό παράδειγμα πρότασης που δεν σημαίνει τίποτε απολύτως. Τουλάχιστον όμως είναι εύκολο να καταλάβουμε τι πραγματικά θέλει εκείνη ή εκείνος που τη λέει ("σας χρειάζομαι, ρε 'σείς, ψηφίστε με ξανά ώστε να συνεχίσω να πληρώνομαι από τους φόρους σας").
- Και βεβαίως δεν είναι μόνο οι πολιτικοί που επιδιώκουν να κερδίζουν από τις αερολογίες. Θυμάμαι τώρα περίπτωση γνωστού κάποιου φίλου μου (μετάφραση: δικός μου γνωστός ήταν), ο οποίος ανέκαθεν αγνοούσε τι σημαίνει "ντρέπομαι γι' αυτά που λέω". Όλοι όσοι τύχαινε και τον γνώριζαν, μετά από λίγο καιρό είχαν την ίδια απορία: "Μα καλά, ποια ακριβώς είναι η δουλειά του;". Και είχαν αυτή την απορία γιατί ενώ ήταν φανερό ότι ο τύπος δεν είχε ιδέα για ό,τι προσπαθούσε να πείσει ότι τον απασχολεί (τεχνολογία, τάσεις, πού πάνε τα πράγματα κ.λπ.), την ίδια στιγμή κατείχε πολύ καλή θέση στην Εταιρεία (ΤΜ) και μάλιστα ζηλευτό μισθό (ισχυρίζονταν οι φίλοι εκείνου του φίλου μου).

- Αερολογίες και κανένα νόημα στην πολιτική, λοιπόν, αλλά το ίδιο και σε άλλους τομείς: Τεχνολογία, Οικονομία, Τέχνη, Δημόσιες Σχέσεις κ.λπ. Σε κάθε περίπτωση, το κλειδί είναι να χρησιμοποιείς φράσεις και λέξεις-κλειδιά, οι οποίες ενώ δείχνουν να συμπυκνώνουν πολύ και βαθύ νόημα στην πραγματικότητα είναι λαμπερά περιτυλίγματα με ωραίες κορδέλες που συσκευάζουν τίποτα απολύτως.
- Ασχέτως πάντως από τις απόπειρες άλλων να μας εξαπατήσουν με αερολογίες (και λαϊκισμό), υποψιάζομαι ότι συχνότερα απ' όσο θα θέλαμε να το παραδεχτούμε κινδυνεύουμε από τον ίδιο μας τον εαυτό.
- Εντάξει, όχι όλοι μας, αλλά σίγουρα **κάποιοι** από εμάς (του γράφοντα συμπεριλαμβανομένου).
- Τι εννοώ; Ας ξεκινήσω με ένα απλό παράδειγμα: "γνωρίζω Γερμανικά επιπέδου Oberstufe" (σαν να λέμε "γνωρίζω Αγγλικά επιπέδου Proficiency"). Ασχέτως αν έχω ή όχι το ανάλογο πτυχίο, λέω στον εαυτό μου αυτό το πράγμα. Γιατί το λέω; Μα, γιατί έχω πείσει τον εαυτό μου ότι μπορώ να το πω. Και γιατί έχω πείσει τον εαυτό μου ότι μπορώ να το πω; Χμ, ίσως επειδή συνομιλώ και συνεννοούμαι με Γερμανούς στα Γερμανικά **χωρίς** να έχω πρόβλημα. Επίσης, κανείς δεν παραπονέθηκε για τα Γερμανικά μου, κανείς δεν με πείραξε για την προφορά μου, κανείς δεν επιχείρησε να με διορθώσει για κάτι που είπα κ.ο.κ. Μιλάω Γερμανικά, καταλαβαίνω όταν ακούω Γερμανικά, συνεργάζομαι με Γερμανούς κ.λπ. κ.λπ., επομένως μπορώ και λέω ότι γνωρίζω Γερμανικά επιπέδου Oberstufe.
- Μήπως όμως κοροϊδεύω τον εαυτό μου; Μήπως οι Γερμανοί με τους οποίους συνεργάζομαι ή έχω πάρε-δώσε, είναι αρκετά ευγενικοί για να με πειράζουν ή να με διορθώνουν; Μπορεί οι άνθρωποι απλά να με καταλαβαίνουν κι αυτό να τους αρκεί. Ίσως δεν αισθάνονται τόσο φίλοι μου, ώστε να παίρνουν το θάρρος για να μου κάνουν υποδείξεις περί γραμματικής ή έστω να μου λένε ότι την τάδε φράση κανείς δεν τη χρησιμοποιεί.
- Τελικά τι ισχύει; Ξέρω Γερμανικά επιπέδου Oberstufe ή όχι;
- Προσωπικά τα Γερμανικά μου είναι επιπέδου "Καλημέρα κυρία Μάιερ, πώς είστε;" και μέχρι εκεί, αλλά δεν είναι αυτό το θέμα μας. Το θέμα μας είναι ότι μερικές φορές **νομίζουμε** ότι ξέρουμε κάτι, στην πραγματικότητα όμως βρισκόμαστε ασυμπτωτικά κοντά στο "ο τύπος δεν έχει ιδέα".
- Σας φαίνεται υπερβολικό όλο αυτό; Αν ναι ίσως φταίει το παράδειγμα που επέλεξα, με τα Γερμανικά. Ξεχάστε λοιπόν τη συγκεκριμένη γλώσσα και σκεφτείτε τα Αγγλικά. Ερώτηση: Σας έτυχε ποτέ να συναντήσετε κάποιον που λέει ότι ξέρει πολύ καλά Αγγλικά, αλλά όταν τον ακούσατε να μιλάει

το επίπεδό του ήταν χειρότερο από το "φάγαμε την καμήλα και μας έμεινε η ουρά αναμονής"; Προσέξτε: Τύποι σαν αυτόν, έστω και με τα **χίλια ζόρια** συνεννοούνται εις την Αγγλική. Ίσως στην αρχή έχουν αμφιβολίες για τις ικανότητές του, δοθέντος όμως ικανού χρόνου που διαφέρει ανά περίπτωση **ξεχνάνε** ότι δυσκολεύονται με τα Αγγλικά και πιστεύουν ότι **ξέρουν** Αγγλικά. Μήπως έχουν πράγματι μάθει; Ίσως ναι, ίσως όμως και όχι.

- Ας έλθω και λίγο στα δικά μας. Σε μια παρέα, κάπου σε μια ωραία καφετέρια, γνωρίζετε έναν ευγενικό νέο. Μετά από δεκαεπτά περίπου λεπτά, ο ίδιος νέος εντέχνως λανσάρει τον εαυτό του ως Τιτάνα του προγραμματισμού. Για την ακρίβεια, δεν λέει ότι είναι προγραμματιστής. Αντίθετα, προτιμά τον όρο "ντιβέλοπερ" και δεν ξεχνά ν' αναφέρει ότι γράφει κώδικα από τα δεκαπέντε του. Εσείς τώρα δεν τον κόβετε για πάνω από 24 κι ενστικτωδώς σκεπτόσαστε να ρωτήσετε, την τελευταία στιγμή όμως ένα βασικότερο ένστικτο (της αυτοσυντήρησης;) πετάγεται και σας λέει: "φακ δατ, ντοντ ίβεν μπόδερ" (ναι, για κάποιο λόγο μιλάει αγγλικά).
- Μπορούμε να γνωρίζουμε αν ο νέος μας φίλος είναι πράγματι Τιτάνας του προγραμματισμού; Αν είχαμε μια κάποια ιδέα για τις ιδιότητες ή καλύτερα για τα χαρίσματα ενός Τιτάνα του προγραμματισμού, ίσως θα μπορούσαμε να γνωρίσουμε. Αλλά ξέρετε κάτι; Δεν έχει σημασία τι νομίζουμε / γνωρίζουμε εμείς – γουί ντοντ ίβεν μπόδερ του νόου. Ας δούμε το ερώτημα από τη σκοπιά του φίλου μας, ο οποίος λέει στον εαυτό του ότι είναι πολύ καλός προγραμματιστής.
- Γιατί το λέει αυτό; Γιατί κάποιος, οποιοσδήποτε, θα έλεγε κάτι τέτοιο στον εαυτό του; Μήπως έχει κάποιο σπουδαίο πρότζεκτ να δείξει, π.χ., στο Github; Ίσως έχει, ίσως όμως και να μην έχει. Μήπως επειδή γνωρίζει πολλές γλώσσες προγραμματισμού, ότι κι αν σημαίνει αυτό; (Ξέρω ανθρώπους –και τους σέβομαι γι' αυτό– οι οποίοι όταν βλέπουν στο βιογραφικό ενός υποψηφίου μια παράθεση γλωσσών προγραμματισμού αυτομάτως στέλνουν το βιογραφικό στο /dev/null.) Μήπως επειδή έχει έφεση προς την αλγοριθμική σκέψη; Μήπως επειδή γνωρίζει πώς να επιτίθεται σε ένα δεδομένο πρόβλημα ώστε να παράγει αποδοτικό κώδικα και δεν ενδιαφέρεται για εντυπωσιασμούς;
- Ξέρετε κάτι; Τώρα που το σκέφτομαι κάπως καλύτερα, νομίζω ότι κάποιος που γνωρίζει κάτι καλά, προγραμματισμό, Γερμανικά, μαγειρική, Οικονομικά κ.ο.κ., αποφεύγει να βαθμολογεί τον εαυτό του.
- Κι αποφεύγει να τον βαθμολογεί, ακόμη κι όταν δεν είναι κανείς άλλος εκεί γύρω.

POST CARD

THE ADDRESS TO BE WRITTEN ON THIS SIDE



ονέιλ μέιλ

Αγαπητοί μου,

"Απορώ με το κουράγιο σου", μου λέει σήμερα το πρωί η μητέρα της εξαδέλφης πηγαίνοντας προς την κουζίνα.

"Απορώ πού τη βρίσκεις την όρεξη. Πραγματικά απορώ. Αντί να κάτσεις να χαρείς το Σαββατοκύριακο της Εθνικής Εορτής όπως όλος ο κόσμος, εσύ εκεί, κολλημένος στη γραφομηχανή."

Δεν είπα τίποτα. Κοίταξα για λίγο τη λευκή σελίδα.

"Και να πεις ότι κάτι κατάφερνες□ Τίποτα δεν έχεις καταφέρει. Αναρωτιέμαι αν σε διαβάζει κανείς ή γράφεις μόνο και μόνο για να λες ότι γράφεις σε περιοδικό."

Δεν είχα τι να της απαντήσω. Έριξα μια ματιά στον πάκο με τα παλιότερα τεύχη που είχα κουβαλήσει από το ντουλάπι, μετά ξανακοίταξα τη λευκή σελίδα.

"Να βάλω τουλάχιστον την τηλεόραση να βλέπουμε την παρέλαση ή κι αυτό θα σε ενοχλήσει;", ακούστηκε να λέει χωμένη στην κουζίνα.

Την ερώτησή της ακολούθησαν κάποιες στιγμές απόλυτης σιωπής.

Δεν το είχα συνειδητοποιήσει πως δείχνουν ακόμα την παρέλαση στην τηλεόραση. Νόμιζα πως το είχαν κόψει.

Παρέλαση□ Τι μου θύμισε!

Θυμάμαι πριν από χιλιάδες χρόνια που στην τηλεόραση έδειχναν το καρναβάλι της Πάτρας. Μαζευόμασταν όλη η οικογένεια και παρακολουθούσαμε έκθαμβοι στην ασπρόμαυρη οθόνη τα φωσφοριζέ φαντασμαγορικά άρματα να περνούν αργά αργά.

Παρουσιαστής ήταν πάντα ο Άλκης Στέας.

Εγώ είχα ένα μικρό πλαστικό αποκριάτικο σφυρί με καραμούζα και κάθε φορά που δεν μας κοιτούσε κανείς κοπανούσα στα κρυφά την εξαδέλφη. Κι αυτή βέβαια δεν καθόταν με σταυρωμένα τα χέρια. Αρχίναγε να με τσιγκλάει με το μαρκαδόρο και προσπαθούσε να μου αρπάξει το σφυρί.

Για λίγο γινόταν ένας μικρός χαμός αλλά μετά ηρεμούσαμε ξανά. Βοηθούσε η φωνή του Άλκη Στέα. Πραγματικά, ακόμα τη θυμάμαι την φωνή του. Να περιγράφει με μάλλον δυσανάλογο ενθουσιασμό τα γυαλιστερά αλλά κάπως ξεπλυμένα αποκριάτικα άρματα και να προσπαθεί να καλύψει τα αναπόφευκτα κενά στην αφήγηση, παίρνοντας τυχαίες μικροσυνεντεύξεις από διάφορα σημαίνοντα και ασήμαντα πρόσωπα που περνούσανε από δίπλα του.

Εμείς χαιρόμασταν.

Δεν μας ενοχλούσε τίποτα. Ούτε ο νομάρχης, ούτε οι βουλευτές της περιφέρειας, ούτε οι τοπικοί παράγοντες που σπρώχνονταν κάθε τόσο μπροστά από την κάμερα για να μας πούνε πόσο ευτυχείς ήταν κι εκείνοι που μοιράζονταν μαζί μας μια τόσο κοσμοϊστορική στιγμή.

Αυτές ήταν εποχές, αγαπητοί μου. Όχι όπως σήμερα, που όλοι είναι σκυμμένοι πάνω στα άιΦον και χαϊδολογούν οθονιές πάνω κάτω. Ας πούμε η Μαιρού--

"Είσαι εντάξει; Όλα οκέι;" Η μητέρα της εξαδέλφης, που είναι Ινδουίστρια, ανήσυχη με τη σιωπή μου, είχε ξεπροβάλει στην πόρτα της κουζίνας. Με κοίταξε καχύποπτα. "Το φάρμακό σου το πήρες;"

"Νο πρόμπλεμ", της λέω εγώ και άρχισα να χτυπάω τυχαία τα πλήκτρα της γραφομηχανής.

Αυτή πέρασε από μπροστά μου με κατεύθυνση προς την τηλεόραση.

Την άναψε χωρίς δισταγμό.

Γύρισα κι άρχισα κι εγώ να κοιτάζω. Η παρέλαση είχε φτάσει στο τέλος της. Η κάμερα έδειχνε τον κόσμο που έφευγε σιγά σιγά, τη βροχερή ατμόσφαιρα της Θεσσαλονίκης, και τους επισήμους που περίμεναν τη σειρά τους για να κάνουν δηλώσεις.

Θα μου επιτρέψετε, πιστεύω, να παρεμβάλω στο σημείο αυτό ένα απόσπασμα από το εορταστικό μήνυμα του Προέδρου της Δημοκρατίας. Πληκτρολογώ όπως το ακούω.

"Τα φτερά της ελληνικής Πολεμικής Αεροπορίας στέλνουν παντού το μήνυμα του ανθρωπισμού και της ειρήνης. Στέλνουν όμως σε κάθε ιταμό εισβολέα και το διαχρονικό μήνυμα των προγόνων μας: 'Μολών λαβέ'. Το έθνος μας είναι υπερήφανο για σας. Και η Παναγία, η υπέρμαχος στρατηγός, να σας συνοδεύει στις πτήσεις σας."

Είμαι απολύτως νηφάλιος και διαυγής. Η φαρμακευτική αγωγή έχει πράγματι αρχίσει να αποδίδει. Στεκόμενος στο ύψος των περιστάσεων, συνεχίζω την απομαγνητοφώνηση χωρίς δισταγμό:

"Είναι το διαχρονικό όχι που μετατρέπεται, αυτομάτως, στο επίσης διαχρονικό ναι στην ειρήνη, στη δημοκρατία, στη δικαιοσύνη και στα θεμελιώδη δικαιώματα του ανθρώπου. Ιδίως δε σ' εκείνα που εμπεδώνουν τις αρχές και τις αξίες της κοινωνικής δικαιοσύνης. Χρόνια πολλά."

Έτσι το είπε, όπως σας το γράφω: Διαχρονικό "ναι" στα θεμελιώδη δικαιώματα που εμπεδώνουν τις αρχές της κοινωνικής δικαιοσύνης. Και μετά: Χρόνια πολλά.

Μάλιστα.

Κατ' αρχάς πείτε μου, γιατί αυτό με μπερδεύει. Από πότε ξεκίνησε αυτή η μόδα να λέμε Χρόνια Πολλά με την ευκαιρία των εθνικών επετείων; Ποιος ακριβώς γιορτάζει; Από όσο θυμάμαι εγώ, Χρόνια Πολλά λέγαμε τα Χριστούγεννα, το Πάσχα, στα γενέθλια, στου Αϊ Γιώργη, στης Αγίας Άννης, άντε και στων Κωνσταντίνου κι Ελένης, και αυτό είναι. Στις 28 Οκτωβρίου, γιατί; Από πότε; Πόσα επεισόδια έχω χάσει;



Τέλος πάντων.

Που λέτε, προτού ξεκινήσω να σας γράφω σήμερα είχα αποφασίσει να πραγματοποιήσω την υπόσχεσή

μου --κατά άλλους απειλή-- να επιστρέψω σε κάποιο από όλα εκείνα τα θέματα που έχω κατά καιρούς αφήσει εκκρεμή στις επιστολές μου.

Κατέβασα που λέτε τα παλιά τεύχη από το ντουλάπι, και ξεκίνησα να τα ξεφυλλίζω. Έφερα από δίπλα και τον φορητό της εξαδέλφης με σκοπό να ψάξω στα PDF. Αυτό που είχα κατά νου ήταν να εντοπίσω τα εκκρεμή θέματα και να τα βάλω σε κάποια σειρά. Είχα πάρει κι ένα τετράδιο για να σημειώνω.

Ξεκίνησα από το πρώτο τεύχος κι άρχισα σιγά σιγά να προχωράω.

Εγκατέλειψα την προσπάθεια όταν συνειδητοποίησα ότι είχα ήδη γεμίσει σελίδες ολόκληρες χωρίς καν να έχω φτάσει στα μισά.

Το υποψιαζόμουν, βέβαια, αλλά τώρα που έκανα τη σχετική στατιστική το βλέπω καθαρά: Για κάθε ένα πράγμα που σας έγραφα άφηνα τουλάχιστον άλλα δύο, ίσως και περισσότερα, στα υπόψιν. Θα επανερχόμουν, τάχα μου, και λοιπά και λοιπά. Αλλά μπα. Μάταια. Δεν επανερχόμουν και τα άφηνα να μαζεύονται. Και τώρα δεν ξέρω από πού να πρωταρχίσω.

Τώρα που τα ξανακοιτάω πάντως βλέπω πως τα περισσότερα από αυτά τα ημιτελή θέματα θα μπορούσαν να ταξινομηθούν σε κάποιες βασικές κατηγορίες.

Ας πούμε, το θέμα της αλήθειας: Τι είναι αλήθεια, πώς ορίζεται η αλήθεια, πώς διαπιστώνεται η αλήθεια. Τάρσκι, Χίντικκα, και δεν ξέρω κι 'γώ πόσοι άλλοι.

Το πρόβλημα της ερμηνείας: Ποιος ερμηνεύει, τι. Πώς διαμορφώνονται οι ερμηνείες; Πώς μεταβάλλονται τα ερμηνευτικά πλαίσια; Γιατί γοητεύει η συνωμοσιολογία;

Το ερώτημα της ιστορίας: Ποιο είναι το αντικείμενο της ιστορίας; Τι αποτελεί ιστορικό συμβάν και τι όχι; Σε τι έγκειται η ιστορικότητα των ερμηνειών; Τι είναι η ιστορική εξέλιξη; Υπάρχουν νόμοι της ιστορίας; Χέγκελ, Μαρξ, Τολστόι, Πόλεμος και Ειρήνη, και τα συναφή.

Το ερώτημα της σχέσης θεωρίας και πράξης: Διάκριση επιπέδων και μεταεπιπέδων. Σύγχυση οντοτήτων με περιγραφές οντοτήτων. Τι είναι η νοημοσύνη; Τι είναι η Τεχνητή Νοημοσύνη; Γκουγκλ, γάτα, κομπιούτορες και λοιπά.

Όλα αυτά είναι θέματα για τα οποία έχω μιλήσει και ξαναμιλήσει. Μοιάζουν τόσο διαφορετικά αλλά αν τα βάλεις δίπλα δίπλα στο χαρτί, αρχίζει να διαφαίνεται πως δεν είναι και τόσο διαφορετικά. Παρόλες τις διαφορές τους τα διατρέχει όλα ένα νήμα.

Πάρτε το πρόβλημα της αλήθειας. Αν ανατρέξετε στις σχετικές επιστολές θα δείτε πως η αλήθεια ορίζεται συνήθως με όρους αντιστοιχίας ή αναντιστοιχίας μιας πρότασης/απόφανσης με την πραγματικότητα στην οποία αυτή η πρόταση/απόφανση αναφέρεται. Κι από εκεί είχαμε περάσει στο πρόβλημα γλώσσας/μεταγλώσσας. Λέξη κλειδί: Γλώσσα.

Αντιστοίχως, για το ερώτημα της ερμηνείας. Θα μπορούσαμε να πούμε πως η αφετηρία του βρίσκεται στο ερώτημα της εγκυρότητας (ή μη) των μηχανισμών με τους οποίους μπορούμε να φτιάχνουμε ισοδύναμες (δηλαδή ισαληθείς) αποφάνσεις, μέσα σε μία γλώσσα. Λέξη κλειδί: Γλώσσα και πάλι.

Σονέι μέι

Σνέιλ μέιλ

Το ερώτημα της ιστορίας: μια γενικευμένη περίπτωση του ερωτήματος της ερμηνείας.

Τα διάφορα ερωτήματα για τις σχέσεις μεταξύ θεωρίας και πράξης, επιπέδων-μεταεπιπέδων, οντοτήτων-περιγραφών οντοτήτων κ.λπ.: αυτά δεν αντικατοπτρίζουν παρά την περιπλοκότητα των αυτοαναφορικών συστημάτων αλληλεπίδρασης, στα οποία συμμετέχουν οι άνθρωποι ως δρώντα όντα που όχι μόνο δρουν αλλά επίσης μιλούν για τη δράση τους. Μιλούν. Δηλαδή πάλι γλώσσα.

Είμαι συνοπτικός, το παραδέχομαι, αλλά ο χώρος μου τελειώνει. Άλλωστε νομίζω πως το νήμα είναι εμφανές. Όλο αυτό τον καιρό μιλούσα για τη γλώσσα, και πιο συγκεκριμένα για την ανθρώπινη γλώσσα.

Το διευκρινίζω αυτό διότι γλώσσα από γλώσσα διαφέρει. Άλλο η χορευτική "γλώσσα" που χρησιμοποιούν οι μέλισσες για να πληροφορήσουν τις άλλες μέλισσες στην κυψέλη για το πού βρίσκεται η τροφή, άλλο η γλώσσα του DNA που διέπει την κωδικοποίηση των αμινοξέων, άλλο η γλώσσα Java ή η Visual C , κι άλλο η γλώσσα στην οποία είναι γραμμένη αυτή εδώ η επιστολή.

Διαφέρουν κι όχι μόνο στην μορφή. Διαφέρουν ουσιαστικά.

Για πολλούς το κοινό χαρακτηριστικό όλων, μα όλων των γλωσσών, είναι το ότι επιτρέπουν την επικοινωνία, με άλλα λόγια τη μετάδοση πληροφοριών από έναν πομπό προς έναν αποδέκτη. Αυτοί εκλαμβάνουν, δηλαδή, την επικοινωνία ως την πρωταρχική λειτουργία της γλώσσας και, ακολούθως, χρησιμοποιούν την έννοια της επικοινωνίας για να ορίσουν τι είναι η γλώσσα.

Σα να μας λένε, δηλαδή, πως γλώσσα είναι αυτό που καθιστά δυνατή την επικοινωνία, και ότι επικοινωνία είναι η πρωταρχική λειτουργία της γλώσσας.

Τη βλέπετε την κυκλικότητα, έτσι;

Τώρα να πούμε και του στραβού το δίκιο, υπάρχουν πράγματι γλώσσες που έχουν ως βασική τους λειτουργία την κωδικοποίηση και μετάδοση πληροφοριών, δηλαδή την επικοινωνία: η γλώσσα των μελισσών, η γλώσσα του DNA, η Java, η Visual C και πολλές άλλες.

Θα εκπλαγείτε, υποθέτω, αγαπητοί μου, αλλά αυτό που προσπαθώ να ισχυριστώ εδώ είναι πως όλον αυτό τον καιρό έκανα την προπαρασκευαστική δουλειά για να μπορέσω να το πω: Η ανθρώπινη γλώσσα *δεν* έχει ως βασική της λειτουργία την επικοινωνία.

Να το λοιπόν, το είπα. Θα πρέπει τώρα να το εξηγήσω.

Αν το εξετάσουμε προσεκτικά θα δούμε πως "επικοινωνία" δεν είναι παρά μια δευτερεύουσα έννοια, μια επιφανειακή περιγραφή.

Στην ανθρώπινη γλώσσα δεν υπάρχει ούτε μήνυμα, ούτε κώδικας, ούτε πληροφορία, τουλάχιστον όχι με τον τρόπο που ορίζονται αυτές οι έννοιες στη θεωρία της επικοινωνίας και αξιοποιούνται στην μελέτη της γλώσσας των μελισσών ή του DNA.

Όσον αφορά στις ανθρώπινες γλώσσες, η χρήση τέτοιων όρων είναι καταχρηστική, αποπροσανατολιστική και συσκοτίζει τη συζήτηση παρά την διευκολύνει.

Θα μου επιτρέψετε, όμως, να το αφήσω αυτό για την άλλη φορά. (Surprise!)

Δεν υπεκφεύγω. Απλά το θέμα είναι τεράστιο και οι εννοιολογικές κακοτοπιές πολλές.

Θα επανέλθω, λοιπόν.

Σας ασπάζομαι,
Θείος Ακάκιος

Skill: Intermediate
Tags: AutoYaST, KVM, libvirt, openSUSE, provisioning, QEMU

Ταχύτατη ανάπτυξη VM και guest OS, από τη γραμμή εντολών

Αναλυτικά και βήμα προς βήμα δείχνουμε πώς δημιουργούμε εικονικές μηχανές QEMU/KVM και κατόπιν πώς τους εγκαθιστούμε το νέο openSUSE Leap 42.3, χωρίς να εγκαταλείψουμε το τερματικό ούτε για μια στιγμή.

Θα δούμε σε πολύ λίγο πώς φτιάχνουμε ένα πλήρες VM με το λειτουργικό του ξεκινώντας **από το μηδέν** και δουλεύοντας αποκλειστικά από τη γραμμή εντολών. Το αληθινό μας κίνητρο δεν πηγάζει από μια κάποια επιθυμία για έξυπνη ή έστω πρωτότυπη ανάπτυξη εικονικών μηχανών. Πολύ περισσότερο, αυτό που θέλουμε είναι να δημιουργήσουμε κάτι σαν template VM κι από εκεί πολύ γρήγορα να συγκροτήσουμε ένα ολόκληρο cluster. Το γιατί θα σχεδίαζε κάποιος κάτι παρόμοιο, καθώς και το πώς θα εργαζόταν για να υλοποιήσει το σχέδιό του, αποτελεί θέμα επόμενου άρθρου. Προς το παρόν δουλεύουμε σε ένα PC με openSUSE Leap, στο οποίο προϋπάρχει ήδη η απαραίτητη υποδομή λογισμικού για τη δημιουργία και τη διαχείριση εικονικών μηχανών QEMU/KVM. Σε περίπτωση που δεν έχετε ασχοληθεί ξανά με το θέμα, προτείνουμε να διαβάσετε τη σχετική σειρά άρθρων (https://deltahacker.gr/?p=15388).

Το νέο μας VM θα έχει έναν εικονικό δίσκο μεγέθους 128GiB, στον οποίο θα εγκατασταθεί το λειτουργικό του σύστημα. Ο δίσκος αυτός στην πραγματικότητα θα είναι ένα sparse file τύπου QCOW2, αποθηκευμένο σε συγκεκριμένο κατάλογο του host OS. Για να χρησιμοποιήσουμε και την επίσημη ορολογία στον κόσμο του libvirt (https://libvirt.org), ο δίσκος του VM δεν είναι παρά ένα volume εντός ενός συγκεκριμένου pool (σκεφτείτε: **volume = αρχείο** και **pool = κατάλογος**).

Δημιουργία νέου storage pool

Για λόγους καλής διαχείρισης, ας μη χρησιμοποιήσουμε κάποιο από τα υπάρχοντα pools. Αντίθετα, ας επιστρατεύσουμε το εργαλείο virsh ώστε να φτιάξουμε ένα νέο. Πρώτα όμως χρειαζόμαστε έναν κατάλογο, ο οποίος θα αποτελέσει το νέο μας pool. Από ένα τερματικό πληκτρολογούμε:

```
[userson at siglufjörður] mkdir /mnt/keflavik/pools/delta
```

Σημείωση. Δεν εργαζόμαστε από το λογαριασμό του root αλλά ο χρήστης μας, με username το `userson`, ανήκει στο group ονόματι `libvirt`. Εννοείται εξάλλου ότι ο ίδιος χρήστης έχει και δικαίωμα εγγραφής εντός του καταλόγου `/mnt/keflavik/pools`. Να υπογραμμίσουμε τέλος ότι στη μεταβλητή περιβάλλοντος `LIBVIRT_DEFAULT_URI` έχει τεθεί η τιμή `qemu:///system`. Συγκεκριμένα, στο αρχείο `~/.bashrc` του `userson` υπάρχει αυτή η γραμμή: `export LIBVIRT_DEFAULT_URI=qemu:///system`.

Το ίδιο το pool το **ορίζουμε** πληκτρολογώντας `virsh pool-define-as delta dir - - - - /mnt/keflavik/pools/delta` (ναι, οι τέσσερις παύλες χρειάζονται). Δείτε όλα τα pools που έχουμε στο σύστημά μας:

```
[userson at siglufjörður] virsh pool-list --all
Name                     State      Autostart
-----------------------------------------
default                  active     yes
delta                    inactive   no
ISOs                     active     yes
localVMs                 active     yes
```

Παρατηρήστε ότι στο εργαλείο `virsh` δώσαμε την εντολή `pool-list` και σ' αυτή περάσαμε την παράμετρο `--all`, ώστε στην έξοδο να συμπεριληφθούν τόσο τα ενεργά (active) όσο και τ' ανενεργά (inactive) pools. Το `delta` που μόλις ορίσαμε είναι ανενεργό – κι αυτό καθόλου δεν πρέπει να μας εκπλήσσει. Προκειμένου να το **ενεργοποιήσουμε** πληκτρολογούμε `virsh pool-start delta`. Επαληθεύουμε ότι το νέο pool έχει πράγματι ενεργοποιηθεί ζητώντας τη λίστα με όλα τα pools του συστήματος – αλλά αυτή τη φορά χωρίς να δώσουμε την παράμετρο `--all`:

```
[userson at siglufjörður] virsh pool-list
Name                     State      Autostart
-----------------------------------------
default                  active     yes
delta                    active     no
ISOs                     active     yes
localVMs                 active     yes
```

Τέλεια όλα εκτός από αυτό το Autostart, το οποίο για το `delta` είναι `no` ενώ για όλα τα άλλα pools είναι `yes`. Αυτό σημαίνει ότι το `delta` δεν θα ενεργοποιείται αυτόματα κατά την εκκίνηση του host OS. Υποθέτουμε ότι θα θέλετε να ενεργοποιείται αυτόματα, οπότε δώστε `virsh pool-autostart delta` και βεβαίως επαληθεύστε:

```
[userson at siglufjörður] virsh pool-list
Name                    State      Autostart
-----------------------------------------
default                 active     yes
delta                   active     yes
ISOs                    active     yes
localVMs                active     yes
```

Θέλετε μήπως περισσότερες πληροφορίες για το νέο pool; Αν ναι, απλά γράψτε:

```
[userson at siglufjörður] virsh pool-info delta
Name:           delta
UUID:           60c43540-25eb-4898-858a-0a54476e3328
State:          running
Persistent:     yes
Autostart:      yes
Capacity:       2.00 TiB
Allocation:     32.20 MiB
Available:      2.00 TiB
```

Δημιουργία νέου volume

Τώρα που έχουμε το νέο μας pool, το `delta`, ας φτιάξουμε μέσα σε αυτό ένα volume. Θα το ονομάσουμε `leapzero.qcow2` και θα αποτελέσει τον εικονικό δίσκο, μεγέθους 128GiB, της μηχανής που πρόκειται να υλοποιήσουμε σε λίγο. Στο τερματικό μας γράφουμε: `virsh vol-create-as delta --name leapzero.qcow2 --capacity 128G --format qcow2`. Πώς βλέπουμε όλα τα volumes ενός pool; Έτσι:

```
[userson at siglufjörður] virsh vol-list delta
Name                    Path
-------------------------------------------------------------
leapzero.qcow2          /mnt/keflavik/pools/delta/leapzero.qcow2
```

Ιδού και περισσότερες πληροφορίες για το νέο volume:

```
[userson at siglufjörður] virsh vol-info leapzero.qcow2 delta
Name:           leapzero.qcow2
Type:           file
Capacity:       128.00 GiB
Allocation:     196.00 KiB
```

Παρατηρήστε ότι ενώ το αρχείο εμφανίζεται να έχει μέγεθος 128GiB, στην πραγματικότητα καταλαμβάνει μόλις 196KiB! Πάρα πολύ ωραία, έχουμε ένα ωραιότατο sparse file και είμαστε πλέον σε θέση να υλοποιήσουμε το VM μας.

Αυτόματη κατασκευή VM

Κατ' αρχάς κατεβάστε, αν δεν το έχετε ήδη κατεβάσει, το ISO image του openSUSE Leap 42.3 (https://software.opensuse.org/distributions/leap): αυτό θα είναι το guest OS του VM που πρόκειται να δημιουργήσουμε. Το αρχείο που θα πάρετε βάλτε το στο pool όπου οργανώνετε τα ISO images. Αμέσως μετά τη μεταφορά μην ξεχάσετε να φρεσκάρετε τα περιεχόμενα του pool, π.χ., πληκτρολογώντας κάτι σαν `virsh pool-refresh ISOs`.

Κάθε εικονική μηχανή QEMU/KVM ορίζεται από ένα αρχείο XML, στο οποίο περιγράφονται τα χαρακτηριστικά του VM. Εμείς τώρα θέλουμε να δημιουργήσουμε ένα νέο VM για το openSUSE, οπότε το πρώτο πράγμα που χρειαζόμαστε είναι αυτό το αρχείο XML. Πληκτρολογήστε:

```
[userson at siglufjörður] virt-install \
> --name leapzero --os-variant opensuse42.3 \
> --memory 2048 --cpu host-model-only --vcpus=2 \
> --network bridge=br0,model=virtio \
> --disk /mnt/keflavik/pools/delta/leapzero.qcow2,bus=virtio \
> --cdrom /mnt/keflavik/pools/ISOs/openSUSE-Leap-42.3-DVD-x86_64.iso \
> --print-xml 1 > leapzero.xml
```

(Τα σύμβολα > στα αριστερά δεν τα πληκτρολογούμε εμείς. Τα εμφανίζει το BASH λόγω του \ στα δεξιά της προηγούμενης γραμμής κι εξαιτίας του [Enter] που πατάμε.) Παίρνουμε έτσι ένα αρχείο XML, το `leapzero.xml`, το οποίο περιγράφει μια εικονική μηχανή με 2GiB RAM, δύο επεξεργαστές κι έναν δίσκο. Στη ίδια μηχανή υπάρχει κι ένα εικονικό CDROM drive, στο οποίο είναι προσαρτημένο το ISO για την εγκατάσταση του openSUSE Leap 42.3.

Το full path προς ένα οποιοδήποτε volume. Σίγουρα θα παρατηρήσατε ότι στην προηγούμενη εντολή δώσαμε το full path προς το QCOW2 image που αντιστοιχεί στο δίσκο της μηχανής, καθώς και το πλήρες path προς το ISO image του μέσου

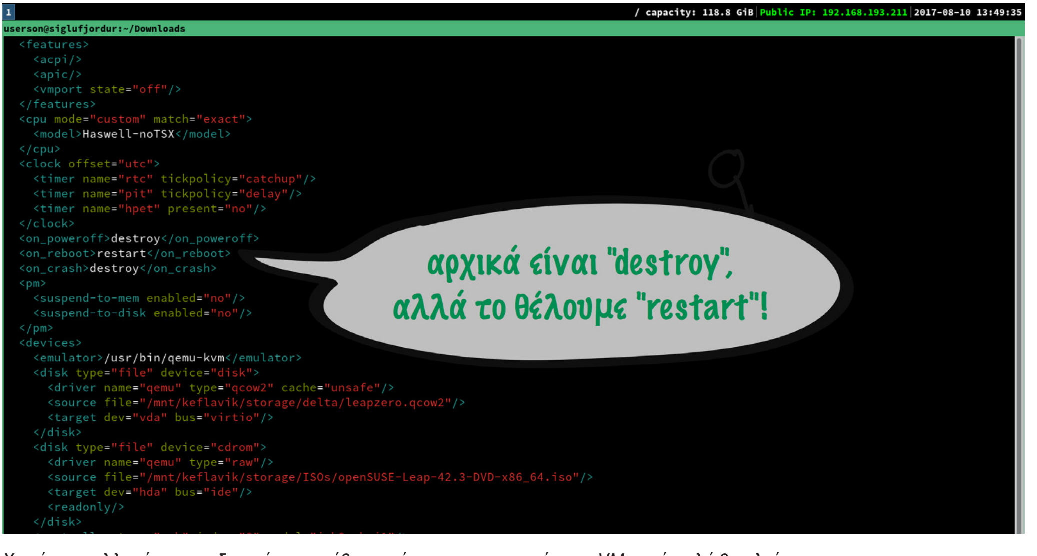
εγκατάστασης. Δεν δώσαμε, μ' άλλα λόγια, τα ονόματα των volumes, ούτε των αντίστοιχων pools. Ίσως θέλετε έναν κάπως αυτοματοποιημένο τρόπο, ώστε με δεδομένα τα ονόματα volume και pool να παίρνετε το full path προς το αρχείο που αντιστοιχεί στο volume. Μια τέτοια μέθοδος θα αποδεικνυόταν χρήσιμη αν, π.χ., σχεδιάζετε ν' αναπτύξετε κάποια scripts, στο πλαίσιο ενός μεγαλύτερου project. Ιδού ένας τρόπος, ο οποίος δεν είναι ούτε ο μοναδικός ούτε ο καλύτερος:

```
virsh vol-dumpxml --vol leapzero.qcow2 delta | grep path | awk -F "<path>"
'{ print $2 }' | awk -F "</path>" '{ print $1 }'
```

(φυσικά, στην περίπτωσή σας αλλάζουν τα ονόματα των volume και pool). Δοκιμάστε τον.

Sound device για το VM; Με την προηγούμενη εντολή, στο XML της μηχανής περιλαμβάνεται και ορισμός για υποσύστημα ήχου. Αν προορίζετε το VM σας για χρήση server, τότε μάλλον δεν χρειαζόσαστε το αντίστοιχο device. Προκειμένου να το αφαιρέσετε ανοίξτε το αρχείο `leapzero.xml` με έναν text editor και διαγράψτε τη γραμμή `<sound model="ich6"/>` (βρείτε τη κοντά στο τέλος του XML).

Πριν συνεχίσουμε ανοίγουμε το αρχείο `leapzero.xml` μ' έναν text editor και ψάχνουμε γι' αυτό το string: `<on_reboot>destroy</on_reboot>`. Αλλάζουμε το destroy σε restart. Το νέο string, δηλαδή, θα είναι αυτό: `<on_reboot>restart</on_reboot>`. Αποθηκεύουμε την αλλαγή, βγαίνουμε από τον editor και είμαστε πλέον βέβαιοι ότι το VM θα συμπεριφέρεται φυσιολογικά κατά την επανεκκίνησή του.



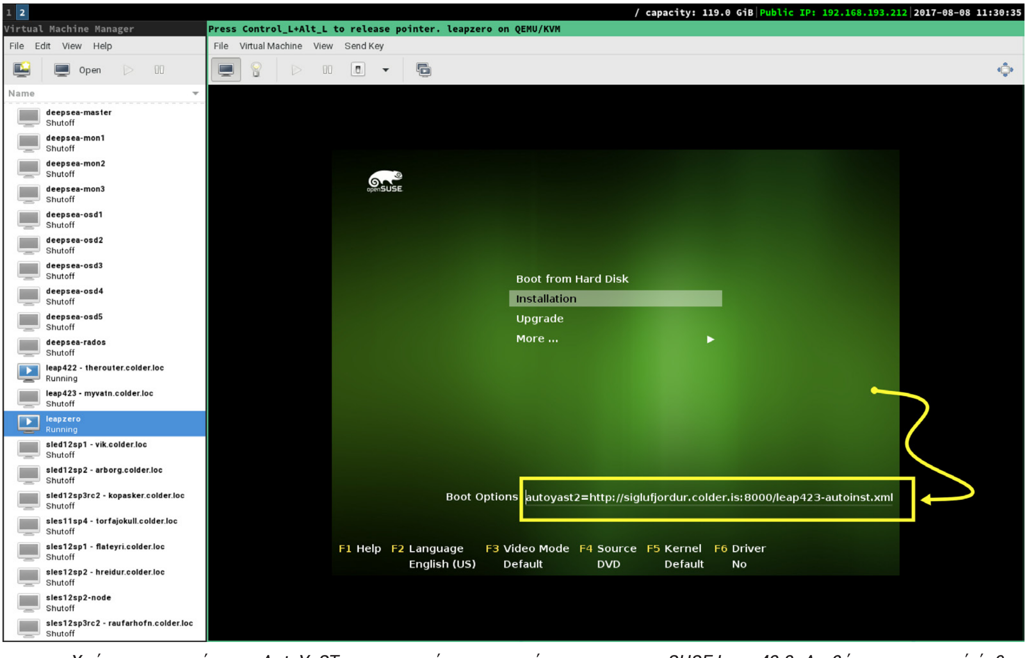
Χωρίς την αλλαγή που επιδεικνύουμε, κάθε φορά που επανεκκινούμε το VM αυτό απλά θα κλείνει.

Θαυμάσια. Έχουμε το ωραιότατο XML της μηχανής όπως ακριβώς το θέλουμε, αλλά όχι την ίδια τη μηχανή. Προκειμένου να την ορίσουμε **με βάση** το XML, αρκεί να πληκτρολογήσουμε `virsh define leapzero.xml --validate`. Η δημιουργία (ορισμός) του VM πραγματοποιείται ακαριαία. Πλέον, δίνοντας `virsh list --all`, μεταξύ των άλλων VMs βλέπουμε κι ένα με όνομα `leapzero`. Η μηχανή συμπεριλαμβάνεται και στη σχετική λίστα του Virtual Machine Manager (δώστε `virt-manager` για να τον ξεκινήσετε).

Το νέο VM δεν έχει ακόμα guest OS. Η εγκατάστασή του γίνεται ξεκινώντας τον εικονικό υπολογιστή, π.χ., με ένα `virsh start leapzero`. Για τη συνέχεια βέβαια θα χρειαστούμε πρόσβαση στην οθόνη (κονσόλα) του. Αφήστε για λίγο το τερματικό και στραφείτε στον Virtual Machine Manager (από το περιβάλλον του, επιλέξτε το VM και κάντε κλικ στο κουμπί [Open]). Μετά την εγκατάσταση του guest OS (δηλαδή του openSUSE Leap 42.3, στο πλαίσιο της παρουσίασής μας) θα μπορέσετε και πάλι να επιστρέψετε στο τερματικό σας.

Αυτόματη εγκατάσταση guest OS με το AutoYaST

Από τη στιγμή που το guest OS για το νέο VM είναι το openSUSE, η ίδια η εγκατάσταση του λειτουργικού συστήματος είναι εύκολο να αυτοματοποιηθεί με τη βοήθεια του AutoYaST (https://deltahacker.gr/autoyast). Το `leap423-autoinst.xml` είναι ένα AutoYaST profile για την εγκατάσταση του Leap 42.3 σε ρόλο server (δεν υπάρχει περιβάλλον γραφικών), στο VM που μόλις φτιάξαμε (κι όχι μόνο). Μπορείτε να κατεβάσετε το προαναφερθέν XML από το https://github.com/colder-is-better/autoyast.



Χρήση του συστήματος AutoYaST για την αυτόματη εγκατάσταση του openSUSE Leap 42.3. Διαβάστε το σχετικό άρθρο (https://deltahacker.gr/autoyast) στο τεύχος 059 (https://deltahacker.gr/deltahacker059) για περισσότερες λεπτομέρειες.

Μετά το πέρας της 100% αυτοματοποιημένης διαδικασίας εγκατάστασης, στο λειτουργικό της μηχανής θα υπάρχει ένας λογαριασμός χρήστη ονόματι User Userson, με username το `userson` και password το `topsecret`. Ο δίσκος του VM, μεγέθους 128GiB, θα έχει δύο κατατμήσεις: μία μεγέθους 2GiB, για χρήση swap, καθώς κι άλλη μία μεγέθους 126GiB, η οποία αποτελεί το root filesystem. Η δεύτερη αυτή κατάτμηση είναι διαμορφωμένη κατά Ext4.

Προετοιμασία template VM

Τώρα που έχουμε το ωραιότατο QEMU/KVM VM με το openSUSE Leap 42.3 εγκατεστημένο, μια καλή ιδέα είναι να το χρησιμοποιούμε ως πρότυπο (template) για τη δημιουργία άλλων VMs. Η βασική ιδέα είναι ότι κλωνοποιούμε το υπάρχον VM όσες φορές θέλουμε κι έτσι δημιουργούμε πολύ γρήγορα νέα VMs. Κάθε νέο VM αρχικά έχει τα χαρακτηριστικά του προτύπου, δηλαδή δίσκο ιδίου μεγέθους, 2GiB μνήμης RAM, μία κάρτα ethernet κ.ο.κ. Από εκεί και πέρα είναι πολύ εύκολο να προσθέτουμε δίσκους, κάρτες δικτύου, ήχου κ.λπ. ή να μεταβάλλουμε την ποσότητα της μνήμης ενός οποιουδήποτε VM. Εργασίες σαν αυτές πραγματοποιούνται με τη βοήθεια του εργαλείου `virsh` και σε άλλα άρθρα θα έχουμε την ευκαιρία να δούμε συγκεκριμένα παραδείγματα. Προς το παρόν, ας προετοιμάσουμε το `leapzero` ώστε να του ταιριάζει καλύτερα ο ρόλος του template VM.

Αφαίρεση συσκευής CD/DVD. Από τη στιγμή που έχουμε τελειώσει με την εγκατάσταση του guest OS, το VM δεν χρειάζεται άλλο τη συσκευή CD/DVD. Αν λοιπόν η μηχανή είναι ενεργή την κλείνουμε, πληκτρολογώντας `virsh shutdown leapzero`. Ακολούθως ζητάμε τη λίστα με τα block devices του VM:

```
[userson at siglufjörður] virsh domblklist leapzero
Target     Source
------------------------------------------------
vda        /mnt/keflavik/pools/delta/leapzero.qcow2
hda        /mnt/keflavik/pools/ISOs/openSUSE-Leap-42.3-DVD-x86_64.iso
```

Ωραία. Το device που δεν χρειαζόμαστε είναι το hda, οπότε το αφαιρούμε αμέσως:

```
[userson at siglufjörður] virsh detach-disk leapzero hda --persistent
```

Επαληθεύουμε ότι πράγματι αφαιρέθηκε:

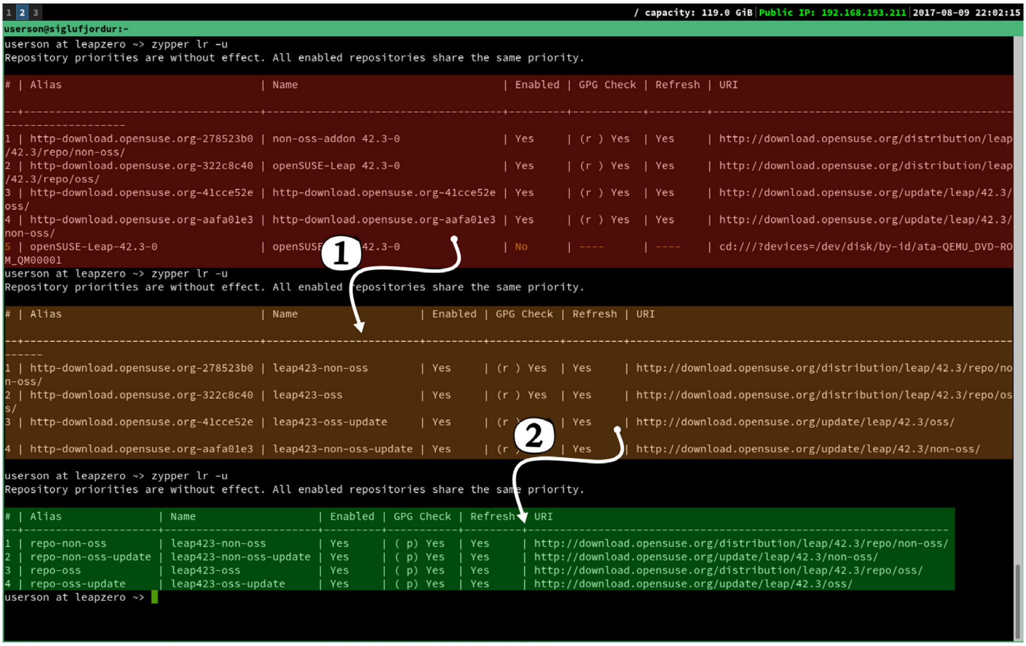
```
[userson at siglufjörður] virsh domblklist leapzero
Target     Source
------------------------------------------------
vda        /mnt/keflavik/pools/delta/leapzero.qcow2
```

Για τη συνέχεια θέλουμε το VM και πάλι ενεργό, οπότε το ξεκινάμε γράφοντας `virsh start leapzero`.

SSH passwordless logins. Το VM είναι για προσωπική χρήση – και πιθανώς το ίδιο θα αληθεύει και για κάθε κλώνο του. Προαιρετικά, λοιπόν, μπορούμε να τοποθετήσουμε το δημόσιο κλειδί του χρήστη μας από τον physical host, στο λογαριασμό ενός χρήστη (π.χ., του `userson`) στο template VM. Κατ' αυτόν τον τρόπο θα συνδεόμαστε στον αντίστοιχο λογαριασμό του VM μέσω SSH και χωρίς να πληκτρολογούμε password (https://deltahacker.gr/?p=13357). Η τοποθέτηση του δημοσίου κλειδιού επιτυγχάνεται γράφοντας κάτι σαν `ssh-copy-id userson@leapzero.colder.is`, όπου στη θέση του `leapzero.colder.is` θα βάλετε το FQDN (Fully Qualified Domain Name) ή απλά την αριθμητική διεύθυνση IP του δικού σας VM.

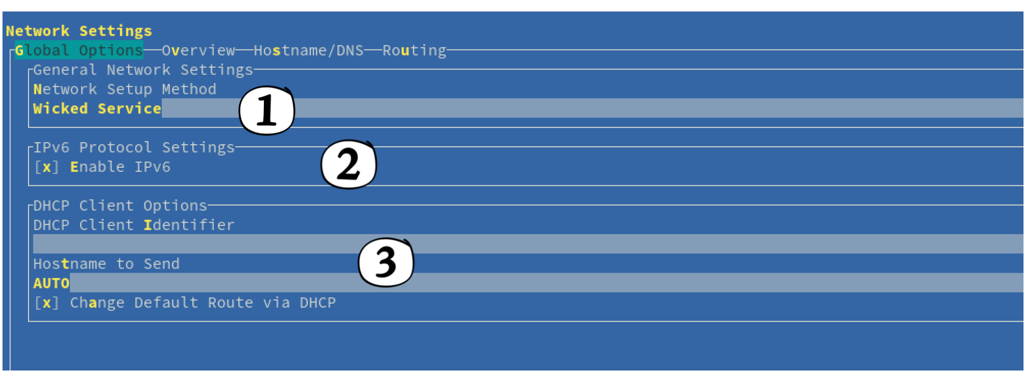
Απλός χρήστης με δικαιώματα passwordless sudo. Αν εγκαταστήσατε το Leap 42.3 χρησιμοποιώντας το AutoYaST profile που σας προτείναμε, τώρα έχετε έναν χρήστη, με username το `userson`, ο οποίος για τις όποιες εργασίες διαχείρισης συστήματος μπορεί και καταφεύγει στο `sudo`. Το password του `userson` είναι ίδιο με το password του `root` – και συγκεκριμένα είναι το `topsecret`. Ίσως θελήσετε να το αλλάξετε, ειδικά αν σε υπηρεσίες του VM έχουν (απομακρυσμένη) πρόσβαση κι άλλα πρόσωπα. Κάτι άλλο που ίσως αξίζει να δείτε, είναι να επιτρέψετε στον `userson` να χρησιμοποιεί το `sudo` χωρίς καν να πληκτρολογεί password (https://deltahacker.gr/?p=17235).

Περιποίηση των repositories. Μετά την εγκατάσταση του λειτουργικού, συνδεθείτε στο VM και πληκτρολογήστε `zypper lr`. Θα δείτε τη λίστα με τα repositories (αποθετήρια) που γνωρίζει ή/και χρησιμοποιεί το λειτουργικό. Ίσως ορισμένα εξ αυτών είναι απενεργοποιημένα, π.χ., επειδή κάποιο repository προέρχεται από το DVD της εγκατάστασης. Άλλα repositories, εξάλλου, ενδεχομένως να έχουν παράξενα aliases ή/και names. Καλό θα ήταν να αφαιρέσετε τα απενεργοποιημένα repositories καθώς και να μετονομάσετε κάποια, αν μη τι άλλο ώστε με μια ματιά να καταλαβαίνετε αμέσως ποιο είναι ποιο. Για τη διαγραφή ενός απενεργοποιημένου repository πληκτρολογήστε κάτι σαν `sudo zypper rr repo_number`, όπου το `repo_number` είναι ο αριθμός του repository όπως εμφανίζεται στην πρώτη από αριστερά στήλη, στην έξοδο της εντολής `zypper lr`. Τώρα, για αλλαγές στα ονόματα ή/και στα aliases των αποθετηρίων, δώστε πρώτα `zypper lr -u`. Χάρη την παράμετρο `-u` παίρνουμε μια νέα στήλη, με τα URI (Uniform Resource Identifier) των repositories. Παρατηρώντας ένα URI κι ανεξάρτητα από τα ονόματα ή τα aliases, είναι εύκολο να συμπεράνουμε, π.χ., αν το τάδε αποθετήριο περιλαμβάνει τα πακέτα OSS, αν το δείνα αποθετήριο έχει τα updates για τα πακέτα που δεν είναι OSS κ.ο.κ. Έτσι, έχουμε όλες τις πληροφορίες προκειμένου οι όποιες αλλαγές σε aliases ή/και σε ονόματα να έχουν νόημα. Για να αλλάξουμε το όνομα του repository υπ' αριθμόν p γράφουμε κάτι σαν `sudo zypper mr -n new_name p`, όπου το `new_name` είναι το νέο όνομα. Για να αλλάξουμε το alias του repository υπ' αριθμόν q πληκτρολογούμε κάτι σαν `sudo zypper nr q new_alias`, όπου το `new_alias` είναι το νέο συνώνυμο.



Περιποίηση της λίστας με τα αποθετήρια λογισμικού. Πρώτα φροντίζουμε για τα ονόματα (1), μετά για τα συνώνυμα (2). Σημείωση: Οι επεμβάσεις καλλωπισμού λαμβάνουν χώρα στο workspace υπ' αριθμόν 3. Ευχαριστούμε.

Φορητή δικτύωση. Πλέον, το VM που έχουμε είναι κατάλληλο να χρησιμοποιηθεί ως πρότυπο, για την ταχύτατη ανάπτυξη άλλων μηχανών. Αξίζει μόνο να βεβαιωθούμε για δυο-τρεις ρυθμίσεις σχετικές με τη δικτύωση. Συνδεόμαστε στο VM μέσω SSH και πληκτρολογούμε `sudo yast2 lan`. Δείτε τα δύο screenshots που ακολουθούν, διαβάστε και τις αντίστοιχες περιγραφές.



Στην εικονική μας μηχανή δεν θέλουμε τον Network Manager να χειρίζεται θέματα δικτύωσης --δεν έχει πολύ νόημα για VMs με ένα WAN interface, χώρια που μας κρύβει ορισμένες παραμέτρους--, γι' αυτό κι έχουμε επιλεγμένη την υπηρεσία Wicked (1). Το IPv6 εξάλλου είναι ενεργοποιημένο, αφού πλέον δεν είναι καθόλου σπάνιο να υποστηρίζεται από --εικονικούς ή φυσικούς-- routers (2). Στα δικά μας εικονικά δίκτυα, τέλος, η απόδοση διευθύνσεων IP και hostnames γίνεται κεντρικά, από τον DHCP server και με βάση τα MAC addresses των client VMs. Γι' αυτό και στο συγκεκριμένο template VM δεν έχουμε αναθέσει τιμή στο DHCP Client Identifier, ενώ για το Hostname to Send έχει τεθεί το AUTO (3).



Το hostname του VM είναι leapzero (1), οποιοδήποτε όνομα όμως προτείνει ο DHCP server γίνεται αποδεκτό (2). Το domain name, εξάλλου, το αφήνουμε κενό, αφού και πάλι θα δεχτούμε το domain name που υπαγορεύει ο DHCP (3).

Σε επόμενα άρθρα θα δούμε τι εννοούμε με την κλωνοποίηση VMs. Μέσα από συγκεκριμένα παραδείγματα θα ξεκινήσουμε από το template VM που μόλις φτιάξαμε και θα αναπτύξουμε πιο εξειδικευμένες μηχανές — έως και ολόκληρα clusters από εικονικές μηχανές.

Skill: Beginner
Tags: OpenVPN, Raspberry, Raspberry Pi, RasPi

Μεταμορφώστε το Raspberry Pi σε OpenVPN server, χωρίς πολλά πολλά

Δεν υπάρχει πλέον καμία δικαιολογία για να παραμένουμε απροστάτευτοι, κάθε φορά που συνδεόμαστε στο Internet από επισφαλή σημεία πρόσβασης. Το ταπεινό αλλά ικανό Raspberry Pi στο σπίτι γίνεται πανεύκολα ικανότατος OpenVPN server, γρήγορα και πολύ εύκολα.

Οι OpenVPN servers είναι μία από τις αδυναμίες μας – και δεν το κρύβουμε. Αναφερόμαστε τόσο σε εκείνους που στήνουμε και ρυθμίζουμε μόνοι μας, όσο και στους έτοιμους που επί πληρωμή παρέχουν εταιρείες όπως η TorGuard (https://torguard.net/aff.php?aff=1790) και η NordVPN (http://nordvpn.com/?ref=1593947de). Στο παρόν άρθρο δείχνουμε πώς χάρη στο πρότζεκτ PiVPN (http://www.pivpn.io) μετατρέπουμε ένα Raspberry Pi με Raspbian (https://deltahacker.gr/raspbian-on-raspi) σε ασφαλή OpenVPN server.

Συνδεόμαστε μέσω SSH στο RasPi και φροντίζουμε ώστε το λειτουργικό σύστημα να είναι πλήρως ενημερωμένο: απλά πληκτρολογούμε `sudo apt update` κι αμέσως μετά `sudo apt upgrade`. Μετά κατεβάζουμε το BASH script από το https://install.pivpn.io. Αποστολή του script είναι να προσθέτει πλήρως λειτουργικούς OpenVPN servers σε εγκαταστάσεις του Raspbian, αναλαμβάνοντας όλες τις σχετικές εργασίες και ρυθμίσεις. Στο χρήστη απευθύνει μερικές μόνον απλές ερωτήσεις, ώστε από τις απαντήσεις να είναι σε θέση να ρυθμίζει σωστά τον υπό εγκατάσταση OpenVPN server. Το εν λόγω script θα το αποθηκεύσουμε στον κατάλογο bin, μέσα στον προσωπικό κατάλογο του χρήστη μας (για εμάς, το username του είναι pi):

```
mkdir -p ~/bin && cd ~/bin
wget https://install.pivpn.io -O pivpn.sh
chmod +x pivpn.sh
```

Ρίξτε μια ματιά μέσα στο script `pivpn.sh`. Μεταξύ άλλων θα παρατηρήσετε ότι οι εργασίες που χρειάζονται δικαιώματα διαχειριστή συστήματος γίνονται με τη βοήθεια του `sudo`. Ο δικός μας χρήστης, ο `pi`, έχει τη δυνατότητα να χρησιμοποιεί το `sudo` χωρίς να πληκτρολογεί το password του `root`. Ο `pi`, μ' άλλα λόγια, έχει δικαιώματα για **passwordless sudo**. Αν θέλετε κι ο δικός σας χρήστης να έχει τα ίδια δικαιώματα, τροποποιήστε το περιεχόμενο του αρχείου `/etc/sudoers` με το εργαλείο visudo. Προσθέστε επίσης και το κατάλληλο αρχείο μέσα στον κατάλογο `/etc/sudoers.d`, ξανά με το `visudo` και με την παράμετρο `-f`. Δείτε πώς τελειώνουν τα περιεχόμενα του δικού μας `/etc/sudoers`...

```
[...]
# User privilege specification
root  ALL=(ALL:ALL) ALL

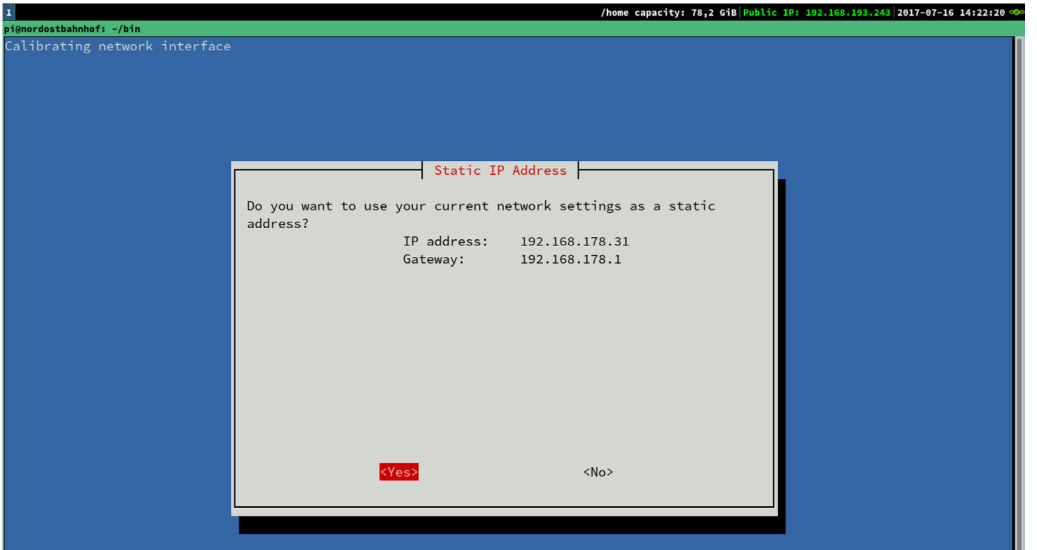
# Allow members of group sudo to execute any command
%sudo ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:
#includedir /etc/sudoers.d
```

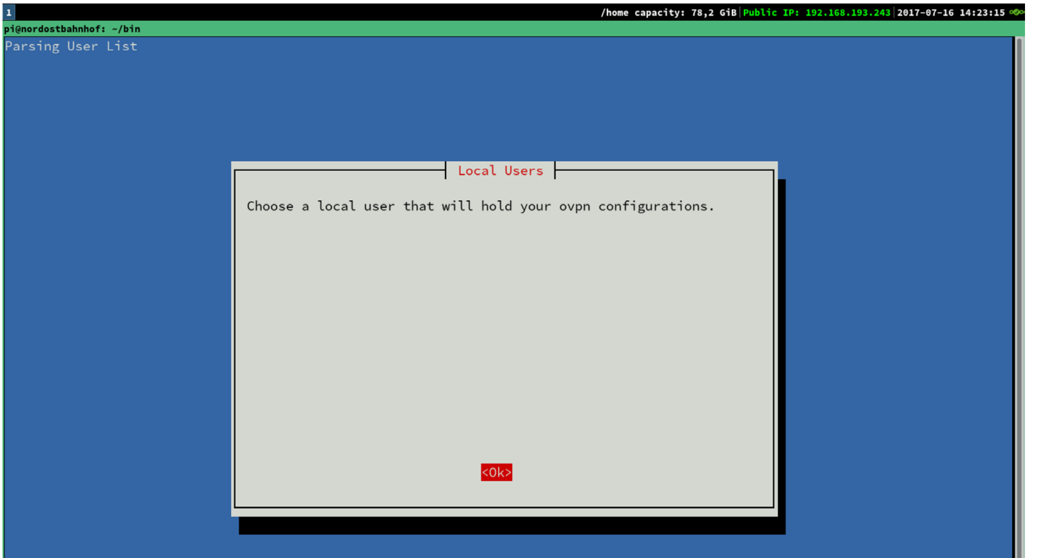
...καθώς και τη μοναδική γραμμή που περιλαμβάνει το αρχείο ονόματι `010_pi-nopasswd`, το οποίο δημιουργήσαμε μέσα στον κατάλογο `/etc/sudoers.d` πληκτρολογώντας `sudo visudo -f /etc/sudoers.d/010_pi-nopasswd`:

```
pi ALL=(ALL) NOPASSWD: ALL
```

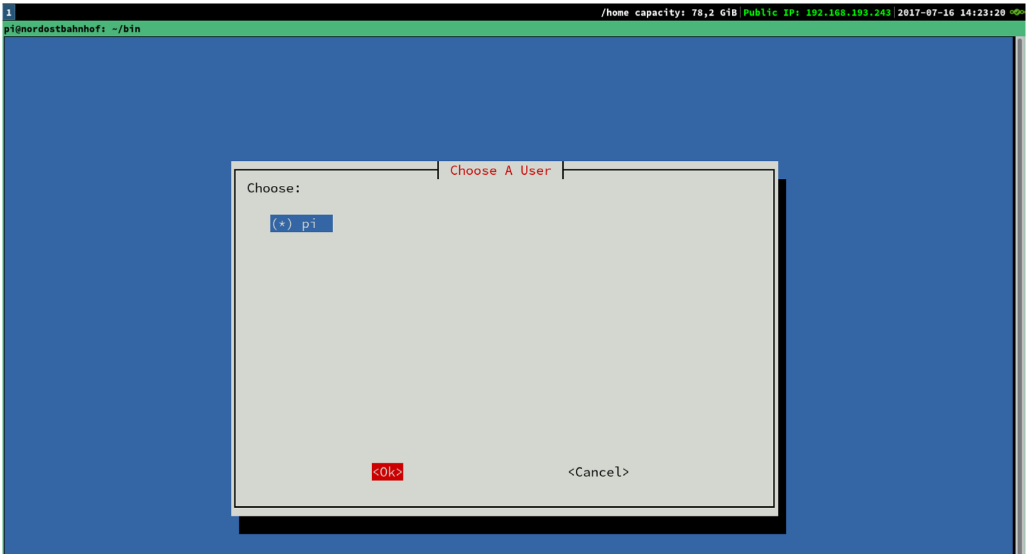
Αλλά ας επιστρέψουμε στο προκείμενο. Βεβαιωνόμαστε ότι βρισκόμαστε μέσα στον κατάλογο όπου αποθηκεύσαμε το script (γράφουμε, π.χ., `cd ~/bin`) και το εκτελούμε δίνοντας `./pivpn.sh`. Δείτε στη συνέχεια μερικά χαρακτηριστικά screenshots, διαβάστε και τις αντίστοιχες περιγραφές.



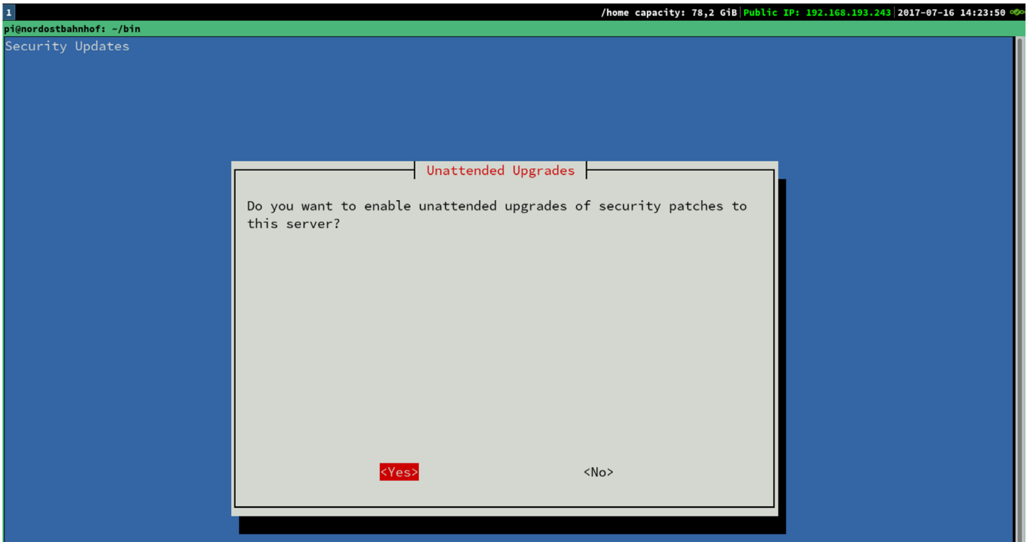
Ο router μπροστά από το Raspberry Pi μας είναι έτσι ρυθμισμένος, ώστε πάντα να του δίνει την ίδια διεύθυνση IP. Αν δεν ίσχυε κάτι τέτοιο, θα έπρεπε να ρυθμίσουμε από μόνοι μας το RasPi για static IP. Σε κάθε περίπτωση, το pivpn ζητά επιβεβαίωση για το IP καθώς και για τη διεύθυνση του gateway, δηλαδή του τοπικού router.



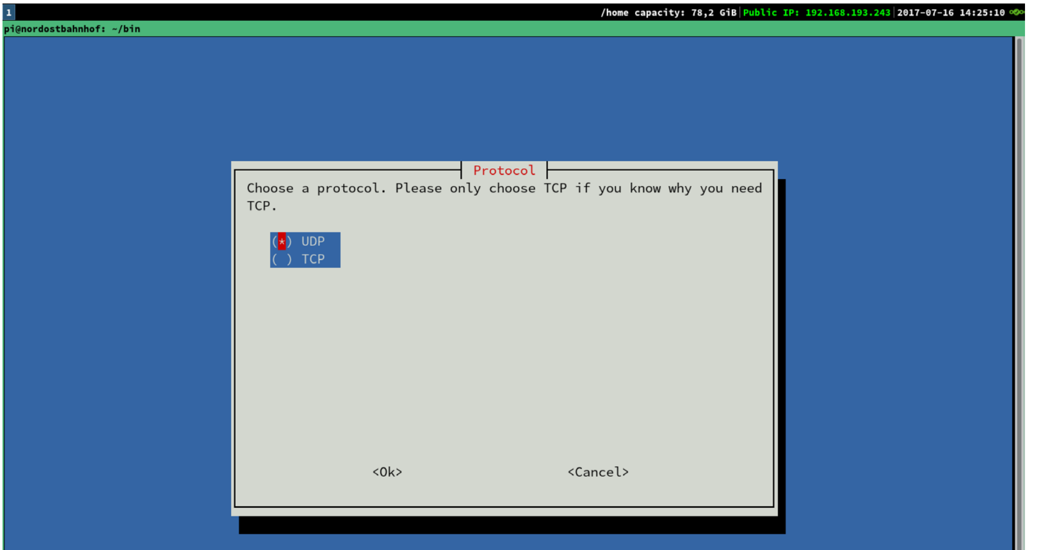
Τα αρχεία σύνδεσης κάθε OpenVPN client θα βρίσκονται αποθηκευμένα και τοπικά, στο RasPi – και συγκεκριμένα στο λογαριασμό ενός απλού χρήστη του συστήματος.



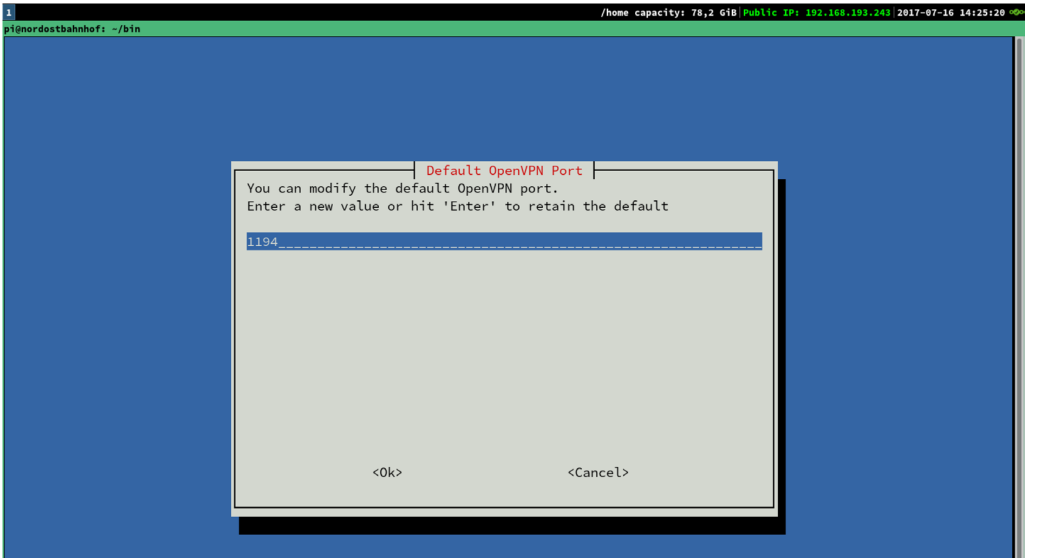
Έναν μόνον χρήστη έχει το σύστημά μας, εκείνον με username το pi, οπότε τώρα δεν έχουμε και πολλές επιλογές.



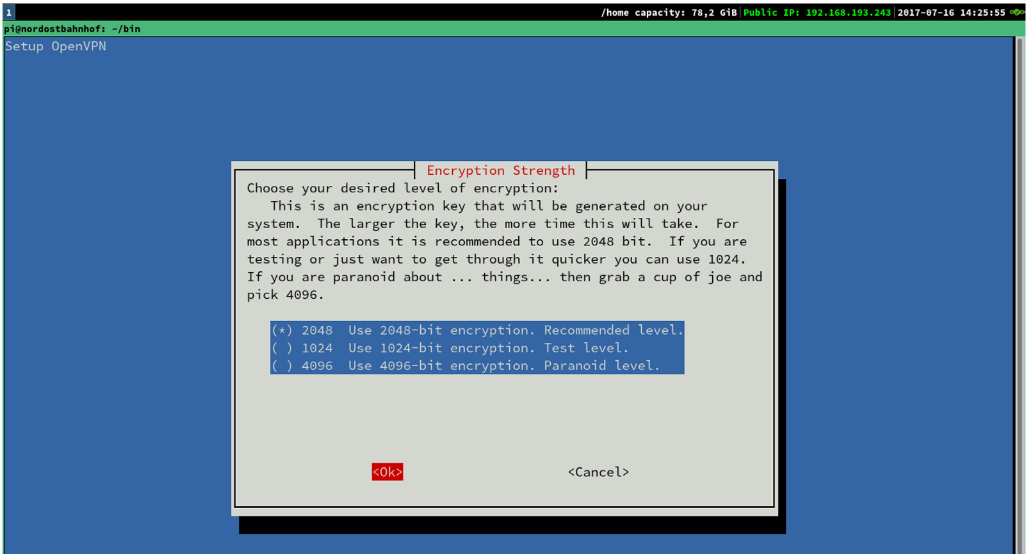
Από τη στιγμή που το Raspberry Pi θα αναλάβει ρόλο server, οι αναβαθμίσεις ασφαλείας καλό είναι να γίνονται χωρίς την εποπτεία μας και φυσικά αυτόματα.



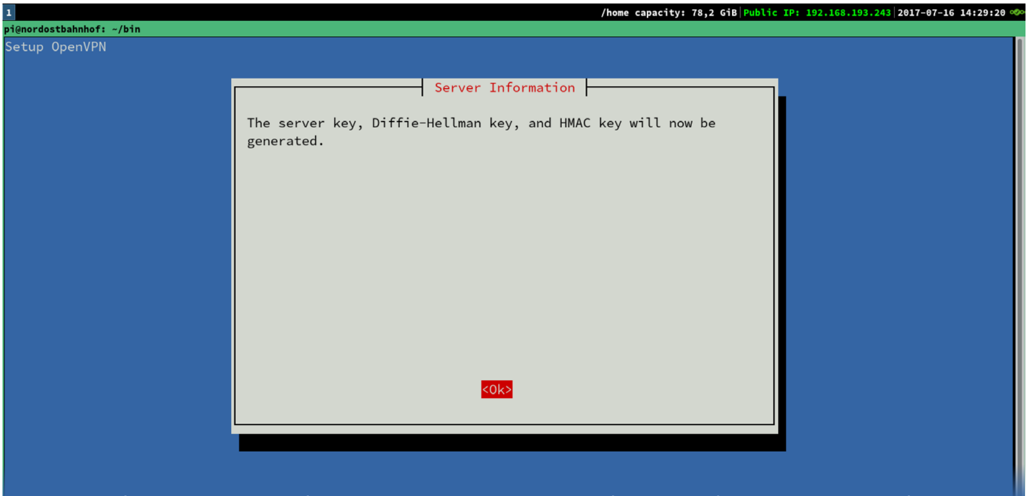
Το πρωτόκολλο μεταφοράς που εξ ορισμού χρησιμοποιεί το OpenVPN είναι το UDP – χωρίς αυτό να σημαίνει ότι είναι αποδεκτή η απώλεια πακέτων μεταξύ client και server. Προτείνεται να αφήσουμε την προεπιλογή. Αν αργότερα διαπιστώσουμε ότι οι συνδέσεις μας προς τον server είναι προβληματικές, μπορούμε να τρέξουμε και πάλι το pivpn.sh κι αυτή τη φορά αντί για UDP να επιλέξουμε το TCP.



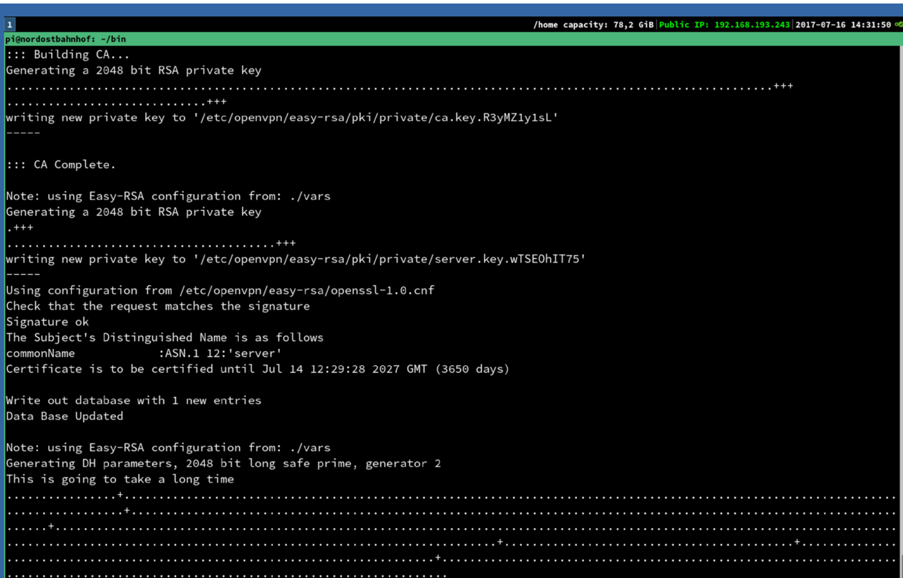
Το προκαθορισμένο port που χρησιμοποιεί το OpenVPN είναι το 1194. Δεν είναι κακή ιδέα να χρησιμοποιήσετε κάποιο άλλο port, απλά έχετε υπόψη ότι ενδεχόμενη αλλαγή δεν σημαίνει και δραματική αναβάθμιση της ασφάλειας του server.



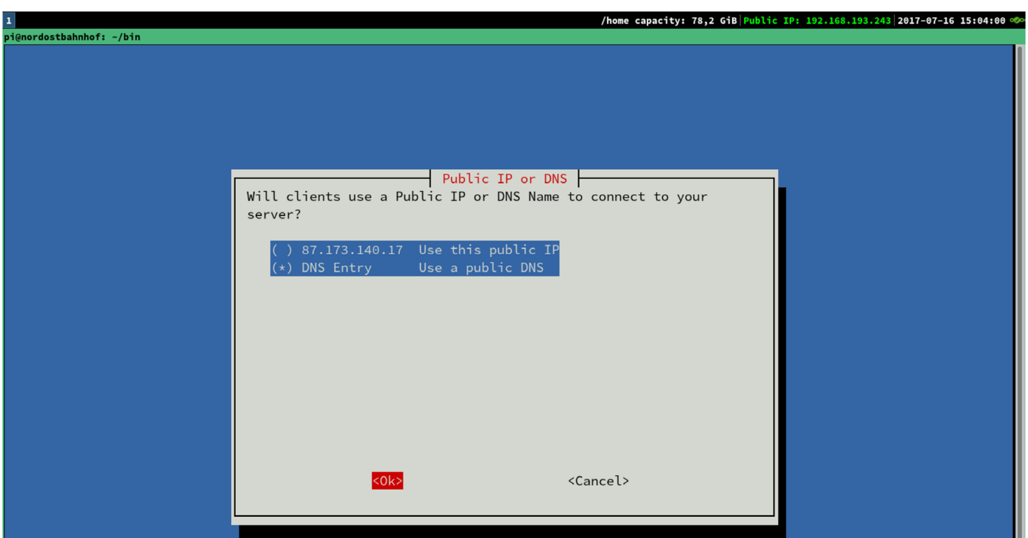
Ώρα για τη δημιουργία του κλειδιού κρυπτογράφησης. Το μήκος του προτείνεται να είναι 2048 bits. Είναι μεγάλος ο πειρασμός να ζητήσουμε την παραγωγή κλειδιού μήκους 4096 bits, αλλά κατά 99,9999999999981276563% δεν πρόκειται να ωφεληθούμε στην πράξη από ένα τέτοιο κλειδί.



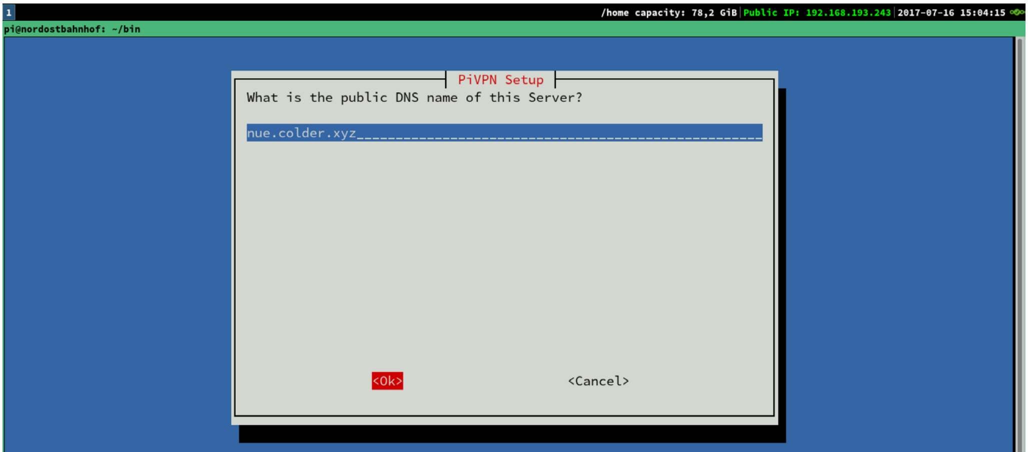
Και μετά την παραγωγή του κλειδιού κρυπτογράφησης, έρχεται η στιγμή για τη δημιουργία μερικών ακόμη κλειδιών και πιστοποιητικών. Παρεμπιπτόντως, αν δεν γνωρίζετε πολλά για τη χρησιμότητα που έχουν τα κλειδιά και τα πιστοποιητικά σε ένα OpenVPN server setup, δεν χρειάζεται ν' ανησυχήσετε. Αποστολή του PiVPN είναι να κρύψει από το χρήστη όλη την πολυπλοκότητα και τις τεχνικές λεπτομέρειες που αφορούν στη λειτουργία του OpenVPN και να του δώσει έναν πλήρως λειτουργικό server.



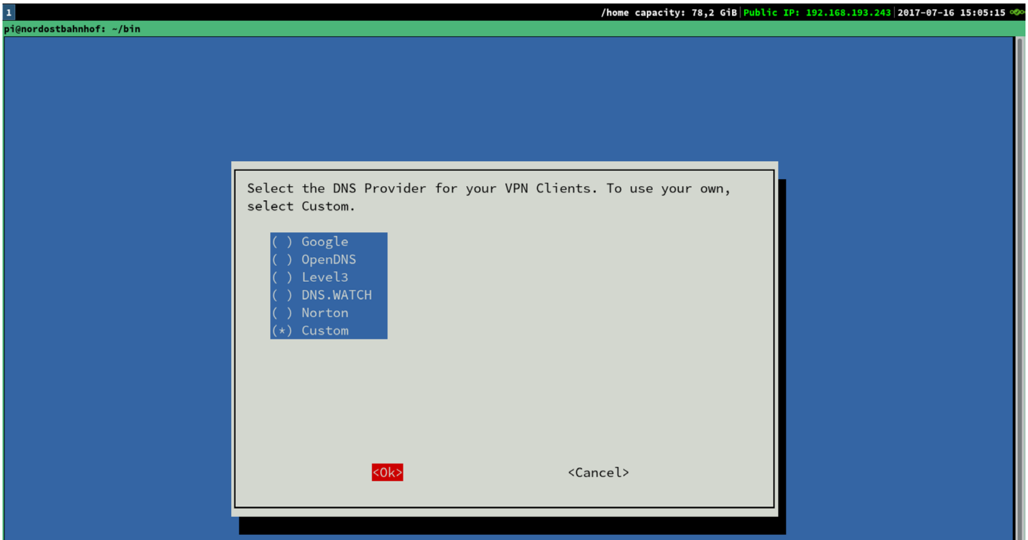
Έχετε κατά νου ότι η παραγωγή των παραμέτρων Diffie-Hellman στο ταπεινό RasPi χρειάζεται αρκετό χρόνο. Δεν πρόκειται για σχήμα λόγου: Στη δική μας περίπτωση πέρασε ένα τέταρτο της ώρας και μετά απλά σηκωθήκαμε και καταπιαστήκαμε με συγυρίσματα στο σπίτι (που για μέρες καταφέρναμε κι αναβάλαμε).



Οι clients του OpenVPN χρειάζεται να γνωρίζουν τη δημόσια διεύθυνση IP του router μπροστά από το RasPi, μ' άλλα λόγια τη διεύθυνση IP που έχει αποδοθεί στη συσκευή από τον ISP. Βέβαια ακόμη κι αν η εν λόγω διεύθυνση είναι δυναμική (το πιθανότερο) κι όχι στατική, καθόλου δεν αποκλείεται να χρησιμοποιείτε κάποια υπηρεσία dynamic DNS. Επιλέξτε, λοιπόν, αναλόγως της περίπτωσής σας.

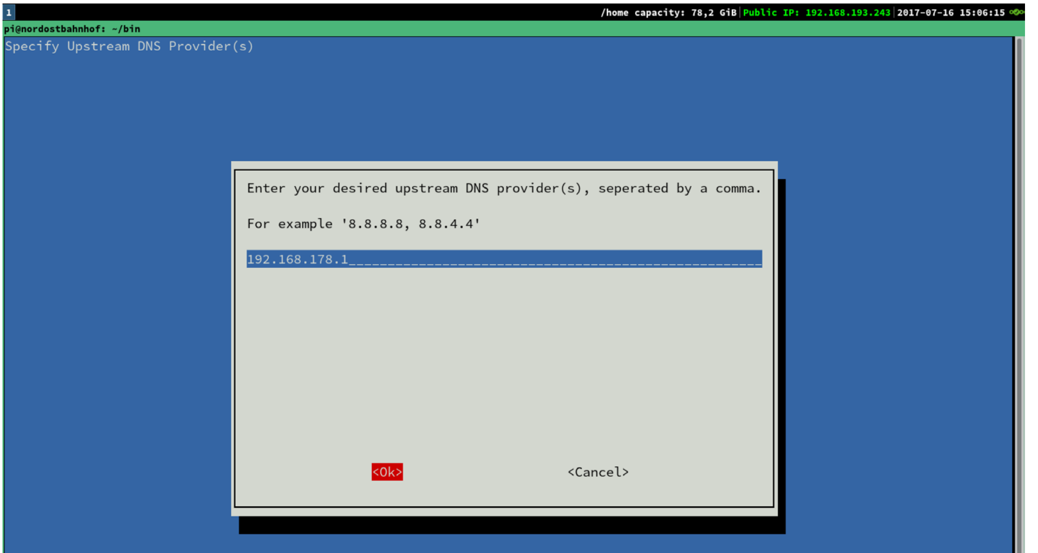


Τυχαίνει και διαχειριζόμαστε τον δικό μας nameserver για το domain όνοματι colder.xyz, ο οποίος παρέχει και API για εφαρμογές που χρησιμοποιούν τις υπηρεσίες του. Έτσι, είναι πολύ εύκολο να διατηρούμε μια αντιστοίχιση μεταξύ δημόσιου IP του router μας και του (sub)domain με όνομα nue.colder.xyz. Αφού φροντίσουμε για τον κατάλληλο κανόνα port forwarding στον router, με το όνομα nue.colder.xyz θα είναι προσβάσιμος κι ο OpenVPN server στο RasPi.

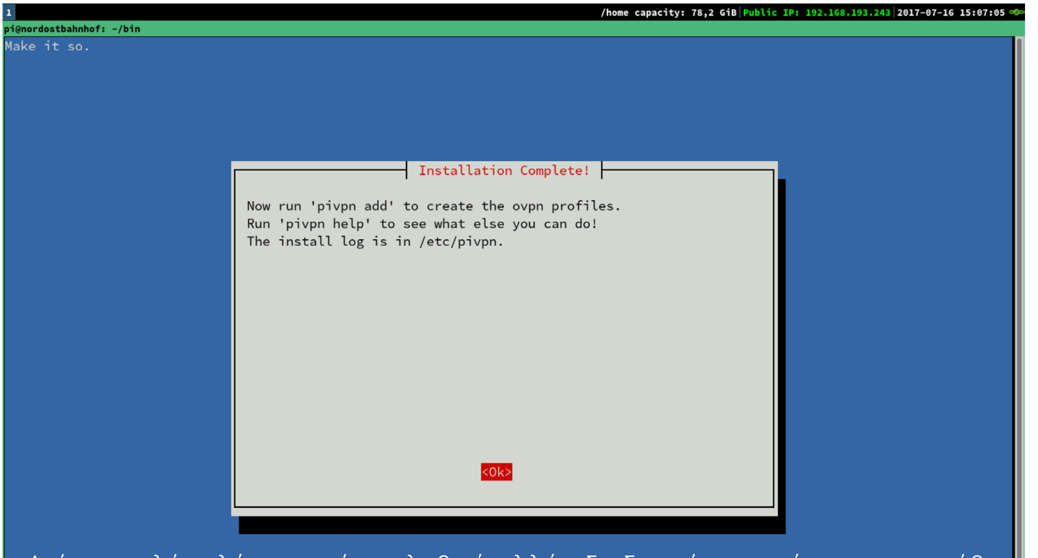


Οι OpenVPN clients θα πρέπει να έχουν τον δικό τους nameserver. Ουσιαστικά, εδώ έχουμε δύο επιλογές:

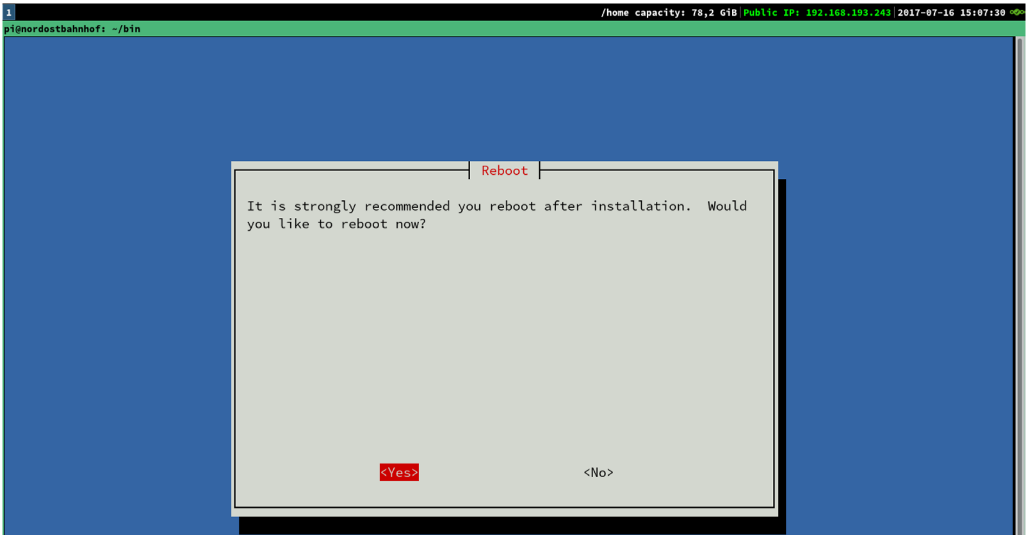
- να καταφύγουμε στους δημόσιους nameservers που παρέχουν εταιρείες όπως Google, OpenDNS κ.ά.
- να υποδείξουμε εμείς κάποιον nameserver, π.χ., αυτόν του router μπροστά από το RasPi.



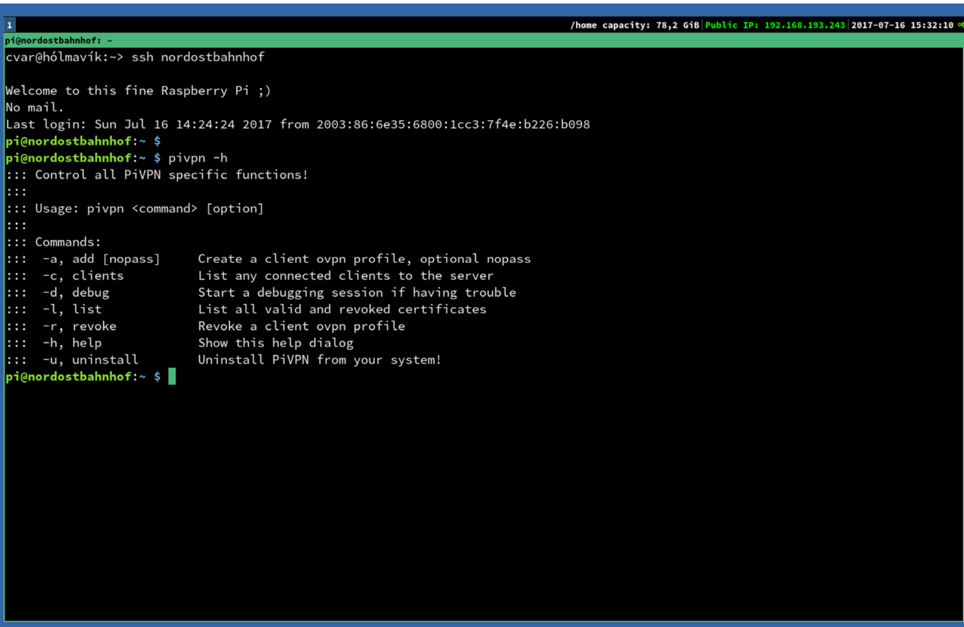
Ως nameserver στη δική μας περίπτωση επιλέξαμε εκείνον που έχει ο router μπροστά από το RasPi: γι' αυτό και πληκτρολογήσαμε το IP του λεγόμενου gateway. Κάναμε τη συγκεκριμένη επιλογή διότι μέσω OpenVPN θέλουμε να συνδεόμαστε στο οικιακό τοπικό δίκτυο και πολύ εύκολα να φτάνουμε σε άλλα μηχανήματα του τοπικού δικτύου, δίνοντας μόνο το hostname τους. Βέβαια κατ' αυτόν τον τρόπο ο ISP βλέπει τα DNS queries των OpenVPN clients, αλλά αυτό καθόλου δεν μας ενοχλεί.



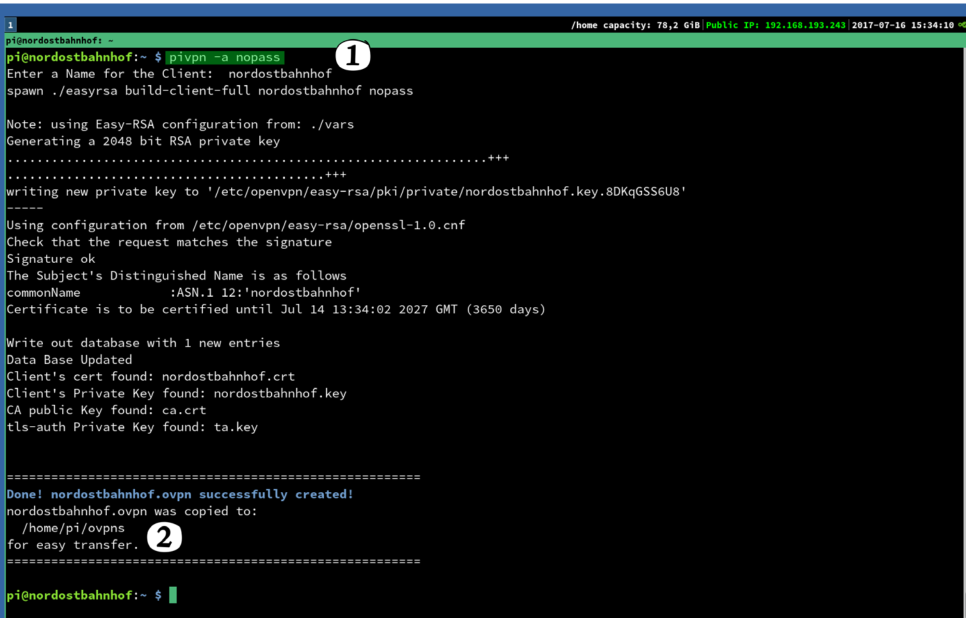
Δείχνει πολύ καλό για να είναι αληθινό, αλλά η διαδικασία εγκατάστασης και ρύθμισης OpenVPN server στο RasPi πράγματι ολοκληρώθηκε επιτυχώς! Για κάθε χρήστη του server πρέπει να δημιουργούμε κι ένα διαφορετικό προφίλ σύνδεσης – κι αυτό επιτυγχάνεται πληκτρολογώντας `pivpn add`.



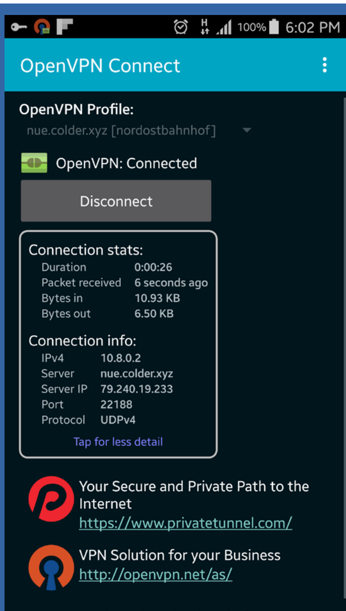
Πριν πάντως κάνουμε οτιδήποτε άλλο με τον OpenVPN server μας, το pivpn.sh μάς προτείνει να επανεκκινήσουμε το σύστημα. Είναι φρόνιμο να μην αγνοήσουμε την προτροπή του.



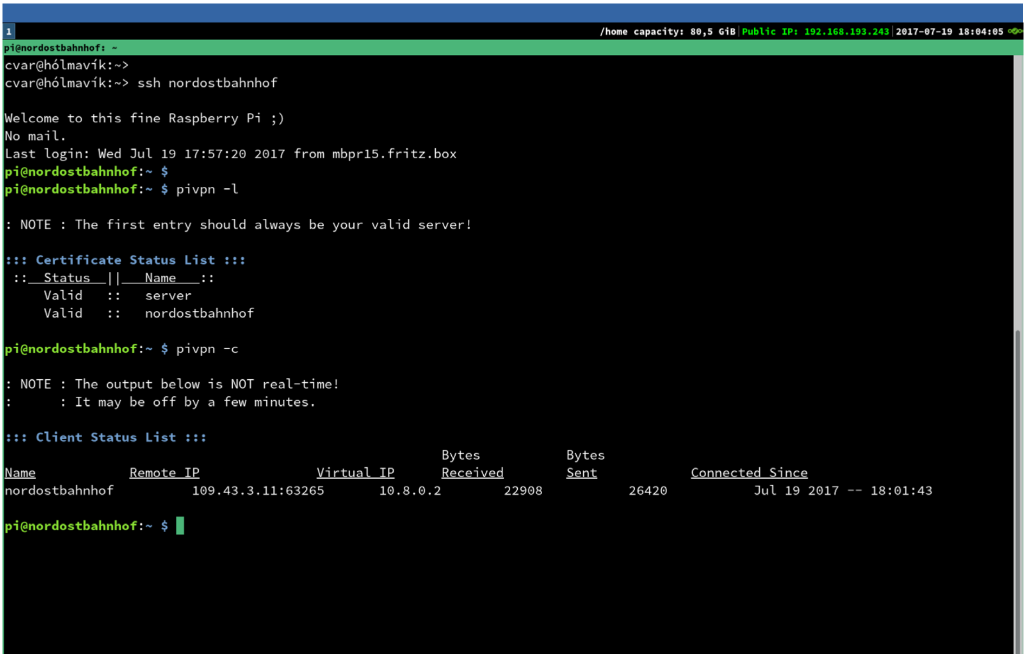
Πληκτρολογώντας pivpn -h βλέπουμε όλες τις παραμέτρους που δέχεται το script, καθώς και τη χρησιμότητα καθεμίας. Θα συμφωνήσετε ότι, εδώ που είμαστε, ιδιαίτερο ενδιαφέρον έχει η -a, για τη δημιουργία προφίλ σύνδεσης πελατών.



Στο παράδειγμα δημιουργούμε ένα νέο προφίλ σύνδεσης, με το όνομα nordostbahnhof (1). Ο χρήστης που θα έχει στη διάθεσή του το συγκεκριμένο προφίλ (αρχείο nordostbahnhof.ovpn) θα μπορεί να συνδέεται στον OpenVPN server χωρίς να πληκτρολογεί password (βλ. επιλογή nopass). Το νέο προφίλ σύνδεσης θα αποθηκευτεί κάτω από τον κατάλογο /home/pi/ovpns (2).



Εφαρμογές σύνδεσης στον OpenVPN server μας διατίθενται για όλα τα λειτουργικά συστήματα και πλατφόρμες. Εδώ βλέπουμε μια ενεργή σύνδεση η οποία επετεύχθη μέσω του OpenVPN Connect για Android – και με βάση το προφίλ που μόλις φτιάξαμε. Για συνδέσεις από Linux, εξάλλου, αρκεί να κατεβάσουμε το προφίλ σύνδεσης (το nordostbahnhof.ovpn, στο πλαίσιο του παραδείγματός μας) και να πληκτρολογήσουμε κάτι σαν sudo openvpn --config ~/nordostbahnhof.ovpn. Να σημειώσουμε τέλος ότι ο αγαπημένος μας client για MacOS X και Windows είναι το Viscosity (https://www.sparklabs.com/viscosity). Προτείνουμε να το κατεβάσετε και να το δοκιμάσετε, αφού διατίθεται δωρεάν για 30 ημέρες. Αν σας κάνει, η αγορά του κοστίζει μόλις 9 δολάρια (ή 14 δολάρια, αν το θέλετε για MacOS X και **ταυτόχρονα** για Windows). Ανεξάρτητα του OpenVPN client, ας υπογραμμίσουμε ότι οι συνδέσεις προς τον OpenVPN server είναι δυνατές μόνον όταν βρισκόμαστε **εκτός του τοπικού του δικτύου**. Επίσης, στον router μπροστά από το Raspberry Pi πρέπει να έχει οριστεί **ο κατάλληλος κανόνας port forwarding**. Έτσι, οι πελάτες που επιχειρούν σύνδεση, π.χ., στο port 1194/TCP του WAN interface του router, θα φτάνουν στο ίδιο port του RasPi.



```
1                                  /home capacity: 80,5 GiB | Public IP: 192.168.193.243 | 2017-07-19 18:04:05
pi@nordostbahnhof: ~
cvar@hólmavík:~>
cvar@hólmavík:~> ssh nordostbahnhof

Welcome to this fine Raspberry Pi ;)
No mail.
Last login: Wed Jul 19 17:57:20 2017 from mbpr15.fritz.box
pi@nordostbahnhof:~ $
pi@nordostbahnhof:~ $ pivpn -l

: NOTE : The first entry should always be your valid server!

::: Certificate Status List :::
 ::  Status  ||   Name   ::
     Valid   ::    server
     Valid   ::    nordostbahnhof

pi@nordostbahnhof:~ $ pivpn -c

: NOTE : The output below is NOT real-time!
:      : It may be off by a few minutes.

::: Client Status List :::
                                                        Bytes           Bytes
Name            Remote IP               Virtual IP      Received        Sent            Connected Since
nordostbahnhof          109.43.3.11:63265       10.8.0.2        22908           26420           Jul 19 2017 -- 18:01:43

pi@nordostbahnhof:~ $
```

Ενεργή σύνδεση πελάτη στον OpenVPN server που τρέχει στο Raspberry Pi μας, όπως φαίνεται με τη βοήθεια του εργαλείου pivpn.

Skill: Intermediate
Tags: KVM, QEMU, libvirt, cloning

Κλωνοποίηση QEMU/KVM VMs, σωστά και γρήγορα

Σε προηγούμενο άρθρο δείξαμε πώς δημιουργούμε QEMU/KVM VMs, ξεκινώντας από το μηδέν κι εργαζόμενοι αποκλειστικά από τη γραμμή εντολών. Έφτασε η ώρα να δούμε και μια γρήγορη μέθοδο κλωνοποίησης εικονικών μηχανών, χωρίς φυσικά να αφήσουμε το αγαπημένο (και βολικό) τερματικό.

Στην παρουσίασή μας για τη γρήγορη κι αυτοματοποιημένη ανάπτυξη μηχανών QEMU/KVM (https://deltahacker.gr/?p=17646), μεταξύ άλλων συζητήσαμε και για την προετοιμασία template VMs. Υποθέτουμε πως έχετε διαβάσει το σχετικό άρθρο, καθώς κι ότι είσαστε εξοικειωμένοι με τις σχετικές έννοιες κι εργαλεία. Το όλο setup για τη συνέχεια έχει ως εξής:

- εργαζόμαστε σε έναν physical host με openSUSE Leap (θα μπορούσε να τρέχει άλλη διανομή Linux)
- η απαραίτητη υποδομή λογισμικού για τη δημιουργία και διαχείριση QEMU/KVM VMs, είναι παρούσα
- υπάρχει το template VM ονόματι `leapzero`, με το καινούργιο openSUSE Leap 42.3 εγκατεστημένο (θα μπορούσε να έχει άλλο guest OS)
- ο εικονικός δίσκος του leapzero είναι ένα QCOW2 image, το `leapzero.qcow2`, εντός του pool ονόματι delta (κατάλογος `/mnt/keflavik/pools/delta`).

Ενημέρωση template VM

Από μια κονσόλα τερματικού ελέγχουμε αν το `leapzero` είναι ενεργό, π.χ., πληκτρολογώντας `virsh list`. Αν δεν είναι, το ενεργοποιούμε δίνοντας virsh start leapzero. Συνδεόμαστε μέσω SSH στο VM (π.χ., `ssh userson@leapzero.colder.is`) και φρεσκάρουμε τις πληροφορίες για τα repositories της διανομής (`sudo zypper ref`). Κοιτάμε για τυχόντα updates `zypper lu`) και, αν υπάρχουν, εφαρμόζουμε τα αντίστοιχα patches (`sudo zypper -n patch`). Ίσως αναβαθμιστεί ο πυρήνας, οπότε σε μια τέτοια περίπτωση επανεκκινούμε το VM (`sudo reboot`). Όταν η όλη διαδικασία

της αναβάθμισης ολοκληρωθεί, έχουμε ένα πλήρως ενημερωμένο template VM. Για τη συνέχεια δεν το θέλουμε ενεργοποιημένο, οπότε αν είμαστε συνδεδεμένοι αποσυνδεόμαστε και κατόπιν κλείνουμε τη μηχανή (`virsh shutdown leapzero`).

Κλωνοποίηση δίσκου

Θα κλωνοποιήσουμε **πρώτα** το δίσκο του template VM, ώστε να είμαστε βέβαιοι ότι ο δίσκος του νέου VM θα είναι επίσης sparse file:

```
[userson at siglufjörður] ~> virsh vol-clone \
> --pool delta leapzero.qcow2 aclone.qcow2
```

Στο εργαλείο `virsh` δώσαμε την εντολή `vol-clone` και σ' αυτή περάσαμε:

- το όνομα του pool στο οποίο βρίσκεται το αρχικό volume (delta, μέσα στο ίδιο pool θα δημιουργηθεί και το νέο volume)
- το όνομα του αρχικού volume (`leapzero.qcow2`)
- το όνομα του volume-κλώνου (`aclone.qcow2`)

Η λειτουργία της κλωνοποίησης θα διαρκέσει κάποια δευτερόλεπτα. Ο συνολικός χρόνος εξαρτάται από το είδος και την ταχύτητα του φυσικού δίσκου επί του οποίου βρίσκεται το pool, καθώς κι από το μέγεθος του πρωτότυπου QCOW2 image. Όταν η κλωνοποίηση ολοκληρωθεί θα πάρουμε ένα μήνυμα σαν αυτό: `Vol aclone.qcow2 cloned from leapzero.qcow2`. Όλα καλά, πάμε τώρα να δημιουργήσουμε ένα νέο VM με βάση το `leapzero`.

Κλωνοποίηση μηχανής

Αρκεί να πληκτρολογήσουμε...

```
[userson at siglufjörður] ~> virt-clone -o leapzero -n aclone \
> --preserve-data -f /mnt/keflavik/pools/delta/aclone.qcow2
```

...και σχεδόν ακαριαία θα πάρουμε το μήνυμα `Clone 'aclone'` created successfully. Παρατηρήστε ότι:

- αντί για το εργαλείο `virsh` αυτή τη φορά καταφύγαμε στο `virt-clone`
- η παράμετρος `-o` δέχεται το όνομα της πρωτότυπης μηχανής (`leapzero`)
- η παράμετρος `-n` δέχεται το όνομα της μηχανής-κλώνου (`aclone`, θα μπορούσαμε να είχαμε δώσει οτιδήποτε άλλο)
- το `--preserve-data` σημαίνει ότι δεν θέλουμε να κλωνοποιηθεί κάποιο volume, αλλά μόνο το XML του αρχικού VM
- η παράμετρος `-f` ακολουθείται από το πλήρες path προς το QCOW2 image που θα αποτελέσει το δίσκο της κλωνοποιημένης μηχανής

Κλώνος μεν, μοναδικός δε

Μόλις δημιουργήσαμε τον πρώτο μας κλώνο, με βάση το template VM που έχουμε στη διάθεσή μας. Πριν κάνουμε οτιδήποτε άλλο, ας δούμε μερικές πληροφορίες που αφορούν στο template VM:

```
[userson at siglufjörður] ~> virsh dominfo leapzero
Id:             -
Name:           leapzero
UUID:           3665fef8-c505-4700-b3f1-8a062a3f7d34
OS Type:        hvm
State:          shut off
CPU(s):         2
Max memory:     2097152 KiB
Used memory:    2097152 KiB
Persistent:     yes
Autostart:      disable
Managed save:   no
Security model: apparmor
Security DOI:   0
```

Στην τρίτη γραμμή των αποτελεσμάτων, βλέπουμε την τιμή του UUID (Universally Unique IDentifier, https://en.wikipedia.org/wiki/Universally_unique_identifier) για το template VM. Το UUID είναι μοναδικό για κάθε VM — ή τουλάχιστον **θα πρέπει** να είναι μοναδικό. Πριν λίγο κλωνοποιήσαμε το template VM, οπότε τώρα είναι λογικό ν' αναρωτηθούμε για το UUID του κλώνου. Από τη στιγμή που μιλάμε για κλωνοποίηση, είναι λογικό να υποθέσουμε ότι τα UUIDs των `leapzero` και `aclone` είναι ίδια. Εκτός βέβαια κι αν το `virt-clone` είναι αρκετά έξυπνο και για τους κλώνους φροντίζει και παράγει νέα UUIDs. Είναι όμως τόσο έξυπνο; Δεν χρειάζεται ν' αναρωτιόμαστε:

```
[userson at siglufjörður] ~> virsh dominfo aclone
Id:             -
Name:           aclone
UUID:           fee9aa57-2e2d-489b-ba1e-e317933823d0
OS Type:        hvm
State:          shut off
CPU(s):         2
Max memory:     2097152 KiB
Used memory:    2097152 KiB
```

```
Persistent:     yes
Autostart:      disable
Managed save:   no
Security model: apparmor
Security DOI:   0
```

Πολύ ωραία, είναι **τόσο** έξυπνο. Για την ακρίβεια είναι εξυπνότερο, αφού εκτός από τα UUIDs φροντίζει να παράγει και MAC addresses για τις κάρτες δικτύου. Δείτε, π.χ., το MAC address που έχει η εικονική κάρτα Ethernet του `leapzero`...

```
[userson at siglufjörður] ~> virsh dumpxml leapzero | grep "mac address" | \
> awk -F "<mac address='" '{ print $2 }' | awk -F "'/>" '{ print $1 }'
52:54:00:17:30:60
```

...κι αμέσως μετά το MAC address που έχει η αντίστοιχη κάρτα του `aclone`:

```
[userson at siglufjörður] ~> virsh dumpxml aclone | grep "mac address" | \
> awk -F "<mac address='" '{ print $2 }' | awk -F "'/>" '{ print $1 }'
52:54:00:c5:25:0a
```

Διαπιστώνουμε, λοιπόν, ότι ο κλώνος δεν είναι **ακριβώς** ίδιος με το template VM, αλλά εκεί που **πρέπει** να διαφοροποιείται πράγματι διαφοροποιείται.

Ενεργοποίηση κλώνου κι ένα θεματάκι δικτύωσης

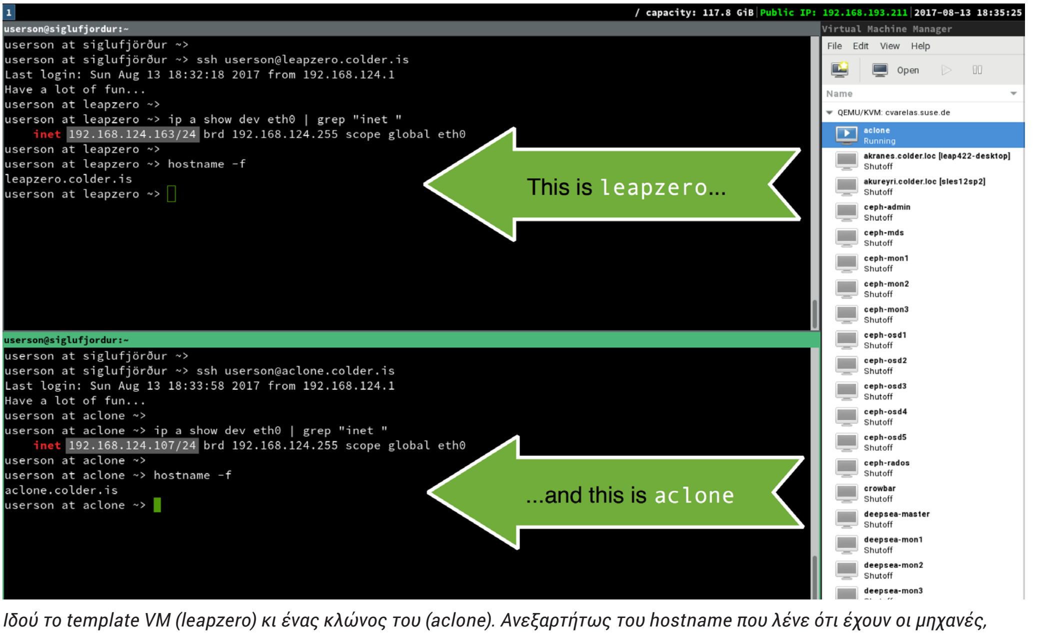
Στο σημείο αυτό μπορούμε να ξεκινήσουμε τον κλώνο, π.χ., δίνοντας `virsh start aclone`. Το template VM περιλαμβάνει το δημόσιο κλειδί του χρήστη μας στον physical host, επομένως το ίδιο ισχύει και για τον κλώνο. Έτσι, μετά από λίγα δευτερόλεπτα θα είμαστε σε θέση για SSH login στο λογαριασμό του `userson` στο `aclone`, χωρίς την πληκτρολόγηση του αντίστοιχου password (το οποίο, παρεμπιπτόντως, σκόπιμα είχαμε φροντίσει ώστε να είναι το πολύ απλό topsecret).

Θα θυμόσαστε εξάλλου ότι το AutoYaST profile που είχαμε χρησιμοποιήσει (https://github.com/colder-is-better/autoyast) για την αυτόματη εγκατάσταση του openSUSE Leap 42.3 στο `leapzero`, θέτει το ίδιο ακριβώς όνομα ως hostname της μηχανής. Η ρύθμιση αυτή θα ισχύει και για τον κλώνο μας. Καταλαβαίνετε, λοιπόν, ότι εδώ έχουμε μια κάπως περίεργη κατάσταση, αφού ένα VM που ονομάζεται `aclone` λέει ότι θέλει ως hostname το `leapzero`. Ακόμα χειρότερα, αν ενεργοποιήσουμε **και** το template VM, τότε θα έχουμε δύο μηχανές που θα λένε ότι θέλουν το ίδιο hostname. Το τι θα συμβεί στην πράξη εξαρτάται από τον (φυσικό ή εικονικό) DHCP server, ο οποίος υπομονετικά κι επιμελώς φροντίζει για τις παραμέτρους δικτύωσης των πελατών του. Κάποιοι DHCP servers εξ ορισμού δίνουν ένα τυχαίο hostname στους πελάτες τους – εκτός κι αν ρυθμιστούν διαφορετικά. Στην

περίπτωσή σας το πιθανότερο είναι ότι έχετε πλήρη πρόσβαση στον DHCP server που εξυπηρετεί τις φυσικές ή/και τις εικονικές σας μηχανές. Γενικά, ζητήματα που αφορούν στη διευθυνσιοδότηση (IPs) κι ονοματολογία (hostnames, domain names) των δικτυωμένων συσκευών, είναι πολύ βολικό να διευθετούνται κεντρικά, από τον ίδιο τον DHCP server. Όσο για τους πελάτες, αυτοί δεν ξεχωρίζουν με βάση τα hostnames που λένε ότι έχουν, αλλά με βάση τα (μοναδικά) MAC addresses των network adapters μέσω των οποίων μιλούν με τον αξιότιμο κύριο DHCP server. Ο δικός μας DHCP server, για παράδειγμα, είναι ο dnsmasq. Στο configuration file έχουμε δύο entries, ώστε τα `leapzero` (MAC address 52:54:00:17:30:60) και `aclone` (MAC address 52:54:00:c5:25:0a) να παίρνουν πάντα τα hostnames που θέλουμε:

```
dhcp-host=52:54:00:17:30:60,leapzero
dhcp-host=52:54:00:c5:25:0a,aclone
```

Προτείνουμε μια παρόμοια διευθέτηση και για τον δικό σας DHCP server. Προαιρετικά, συνδεθείτε στον κλώνο και με τη βοήθεια του YaST αλλάξτε το hostname που η μηχανή γνωστοποιεί στο δίκτυο. Φυσικά, μια τέτοια χειροκίνητη επέμβαση γίνεται γρήγορα άβολη όταν μιλάμε για πολλές μηχανές. Σε τέτοιες περιπτώσεις είναι προτιμότερο να έχουμε καταστρώσει ένα script που δημιουργεί λίστες με ζεύγη της μορφής `<MAC_address, hostname>` όπου το `hostname` θα 'ναι ίδιο με το όνομα της εκάστοτε μηχανής. Αν θέλουμε να πάμε τον αυτοματισμό λίγο παραπέρα, το script θα ενημερώνει αυτόματα τον DHCP server. Είναι φυσικά πολλά αυτά που μπορούμε να κάνουμε, κι όπως υποψιαζόσαστε μόλις δώσαμε αθώα spoilers επόμενων άρθρων.



Ιδού το template VM (leapzero) κι ένας κλώνος του (aclone). Ανεξαρτήτως του hostname που λένε ότι έχουν οι μηχανές, ο DHCP server του τοπικού δικτύου φροντίζει να τους δίνει hostnames βάσει του MAC address καθενός. Οι κανόνες ονοματολογίας έχουν οριστεί από τον διαχειριστή του DHCP server.

Προσθήκη νέου δίσκου στον κλώνο

Ας υποθέσουμε τώρα ότι θέλουμε να χρησιμοποιήσουμε τον κλώνο μας ως host για το Nextcloud. Το αντίστοιχο VM θα χρειαστεί ένα δεύτερο δίσκο, η χωρητικότητα του οποίου αρκεί να είναι στα 512GiB. Δημιουργούμε λοιπόν ένα νέο QCOW2 image, μέσα στο storage pool ονόματι delta:

```
[userson at siglufjörður] ~> virsh vol-create-as delta \
> --name aclone_data.qcow2 --capacity 512G --format qcow2
```

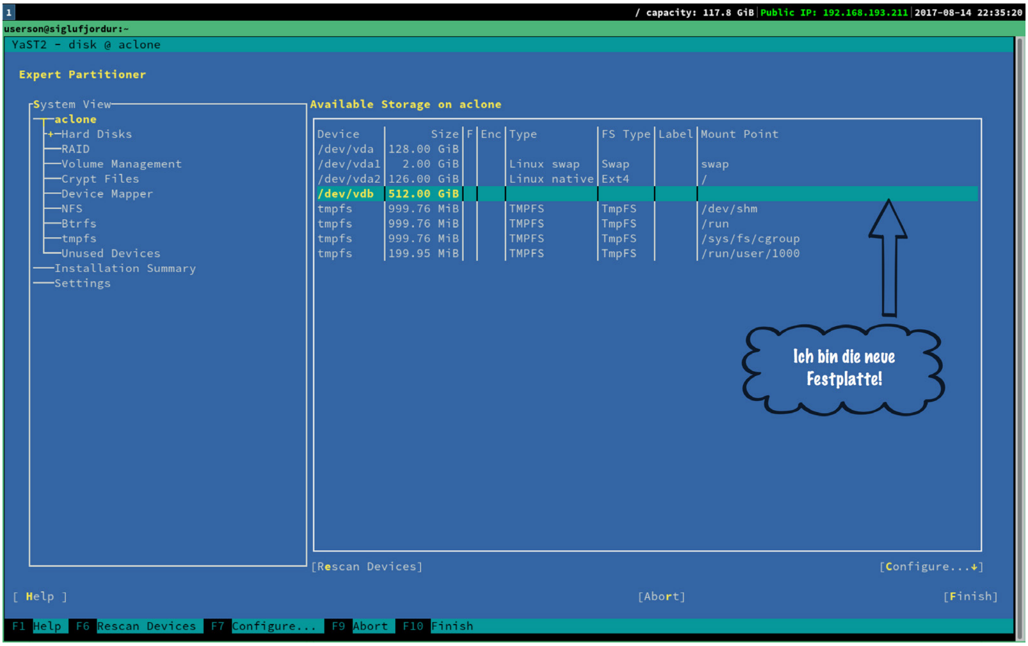
Η δημιουργία του `aclone_data.qcow2` ολοκληρώνεται ακαριαία (μην ξεχνάτε ότι πρόκειται για sparse file). Επαληθεύουμε πως όλα είναι όπως πρέπει:

```
[userson at siglufjörður] ~> virsh vol-list delta
Name                 Path
----------------------------------------------------------------
aclone.qcow2         /mnt/keflavik/pools/delta/aclone.qcow2
aclone_data.qcow2    /mnt/keflavik/pools/delta/aclone_data.qcow2
leapzero.qcow2       /mnt/keflavik/pools/delta/leapzero.qcow2
```

Πολύ ωραία. Ας συνδέσουμε τώρα τον καινούργιο δίσκο στη μηχανή:

```
[userson at siglufjörður] ~> virsh attach-disk aclone \
> /mnt/keflavik/pools/delta/aclone_data.qcow2 vdb \
> --driver qemu --subdriver qcow2 --targetbus virtio --persistent
```

Παρατηρήστε ότι δεν χρειάζεται να κλείσουμε την εικονική μηχανή, προκειμένου να της προσθέσουμε έναν δίσκο. Αφού λοιπόν συνδεθούμε μέσω SSH στον κλώνο και πληκτρολογήσουμε `sudo fdisk -l`, θα διαπιστώσουμε ότι έχει δύο δίσκους: τον `/dev/vda` (αυτός φιλοξενεί το λειτουργικό), καθώς και τον `/dev/vdb` (αυτός είναι ο νέος δίσκος που προσθέσαμε μόλις). Μπορούμε τώρα να δημιουργήσουμε μία ή περισσότερες κατατμήσεις επί του `vdb`, να τις διαμορφώσουμε, να τις προσαρτήσουμε και, γενικά, να χρησιμοποιούμε το δίσκο όπως επιθυμούμε.



Να κι ένας νέος δίσκος για τον κλώνο μας, τον οποίο μάλιστα προσθέσαμε ενώ η μηχανή ήταν ενεργή.

Μεταβολή άλλων χαρακτηριστικών του κλώνου

Αναλόγως της εφαρμογής, η ελαστικότητα των εικονικών μηχανών αποδεικνύεται πολύτιμη. Δείτε για παράδειγμα πόσο εύκολο είναι να διπλασιάσουμε τη μνήμη RAM του κλώνου. Με την προϋπόθεση ότι το `aclone` είναι ανενεργό (powered off), πληκτρολογούμε:

```
[userson at siglufjörður] ~> virsh setmaxmem aclone 4194304 --config
[userson at siglufjörður] ~> virsh setmem aclone 4194304 --config
```

Και μιας και διπλασιάσαμε τη μνήμη, ας διπλασιάσουμε και τον αριθμό των εικονικών επεξεργαστών (από 2 σε 4):

```
[userson at siglufjörður] ~> virsh setvcpus aclone 4 --config --maximum
[userson at siglufjörður] ~> virsh setvcpus aclone 4 --config
```

Ξεκινήστε το aclone, συνδεθείτε μέσω SSH και τρέξτε το top. Παρατηρήστε ότι η μνήμη της μηχανής είναι στα 4GiB (το μέγεθός της εκφράζεται σε KiB, επομένως το σχετικό νούμερο που θα δείτε πάνω αριστερά είναι το `4045656`). Εγκαταλείψτε το top (απλά πιέστε το πλήκτρο [Q]) και πληκτρολογήστε `cat /proc/cpuinfo`. Θα διαπιστώσετε ότι η μηχανή έχε τέσσερις επεξεργαστές (`processor 0` έως και `processor 3`). Τέλος, επαναλάβετε μετά από εμάς: "το virtualization είναι ευλογία".

Skill: Beginner
Tags: Ad-blocking, dnsmasq, Raspberry Pi, RasPi

Μπλοκάρισμα διαφημίσεων για όλο το δίκτυο, με το Raspberry Pi

Δωρεάν ad blockers όπως το uBlock Origin κάνουν εξαιρετική δουλειά, όμως μόνο για τους web browsers που τους χρησιμοποιούν. Σε ένα τυπικό οικιακό δίκτυο έχουμε περισσότερες από μία συσκευές όπου το μπλοκάρισμα διαφημίσεων είναι επιθυμητό, συνεπώς η συνηθισμένη προσέγγιση στο πρόβλημα δείχνει γρήγορα τους περιορισμούς της.

Αναλογιστείτε ότι ακόμη κι αν εφοδιάσουμε με ad blocker **κάθε** web browser σε υπολογιστή, smartphone, tablet –και βεβαίως σε **κάθε** λογαριασμό χρήστη σε **κάθε** μία από τις συσκευές μας–, και πάλι θα έχουμε εφαρμογές, όπως, π.χ., παιχνίδια και apps, οι οποίες θα επιχειρούν να κατεβάζουν τις διαφημίσεις τους. Μία σαφώς πιο αποτελεσματική στρατηγική μεταφέρει τη γραμμή άμυνας από το επίπεδο της εφαρμογής στο επίπεδο ολόκληρου του τοπικού δικτύου – για την ακρίβεια στο επίπεδο του DNS server. Αναλυτικότερα, αν εντός του δικτύου υπάρχει nameserver που μπλοκάρει τα requests προς ad domains, τότε όποια συσκευή χρησιμοποιεί τον εν λόγω nameserver απλά αδυνατεί να φορτώνει διαφημίσεις. Τότε, το μόνο που χρειάζεται να κάνουμε σε κάθε συσκευή είναι μια ρύθμιση του στιλ "στο εξής να χρησιμοποιείς τον Τάδε nameserver", όπου βεβαίως ο "Τάδε nameserver" είναι εκείνος που δεν εξυπηρετεί ad domains. Αντί μάλιστα να ρυθμίσουμε κάθε συσκευή ξεχωριστά, πολύ καλύτερα είναι να ρυθμίσουμε τον DHCP server του δικτύου ώστε στους πελάτες του να προσφέρει αυτόματα αυτόν τον διαφημισιοφάγο-nameserver. (Ναι, πονέσαμε και που το γράψαμε, όμως για κάποιο λόγο δεν μπορέσαμε ν' αντισταθούμε στον πειρασμό.)

Το Pi-hole (https://pi-hole.net) υλοποιεί έναν DNS server για το Raspbian του Raspberry Pi, καθώς και για άλλες διανομές Linux (https://discourse.pi-hole.net/t/hardware-software-requirements/273), ο οποίος απορρίπτει requests προς χιλιάδες ad-domains. Το Pi-hole λειτουργεί επίσης κι ως DHCP server. Έτσι, αν απενεργοποιήσουμε τη σχετική λειτουργία στο υπάρχον modem/router και οι συσκευές μας λαμβάνουν τις παραμέτρους δικτύ-

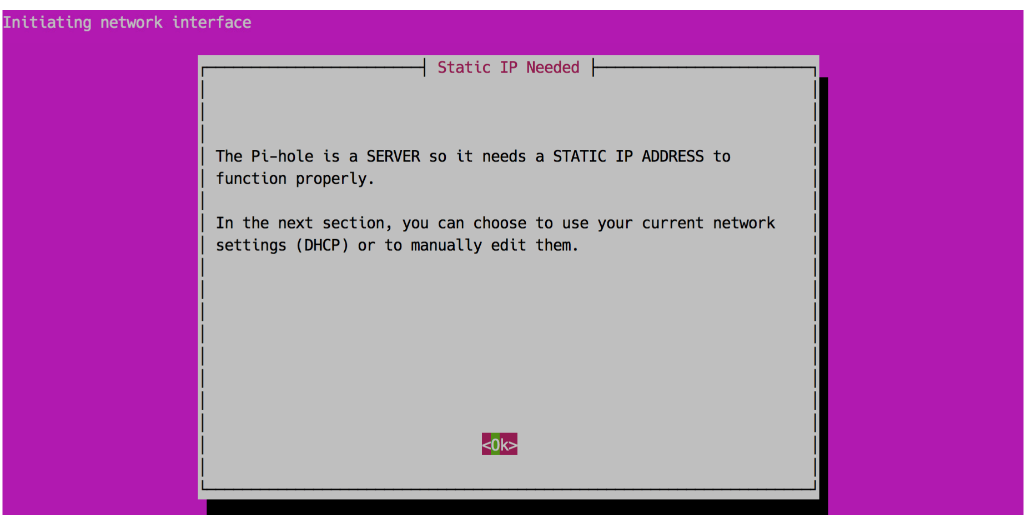
ωσης αυτόματα, τότε όλες τους θα αρχίσουν να χρησιμοποιούν τον nameserver του Pi-hole. Αυτό σημαίνει ad-free εμπειρία στο web, αλλά και καλύτερη αξιοποίηση του διαθέσιμου bandwidth. Αν μάλιστα στο Raspberry Pi έχουμε και το PiVPN (https://deltahacker.gr/raspi-easyvpn), τότε κάθε φορά που είμαστε εκτός τοπικού δικτύου θα έχουμε την επιλογή για σύνδεση στο RasPi μέσω OpenVPN. Εκτός λοιπόν από την ενίσχυση του privacy θα εξοικονομούμε και data, κι αυτό είναι εξαιρετικά επιθυμητό ειδικά όταν συνδεόμαστε στο Internet μέσω του δικτύου κινητής τηλεφωνίας.

Εγκατάσταση, χρήση, πρώτες δοκιμές

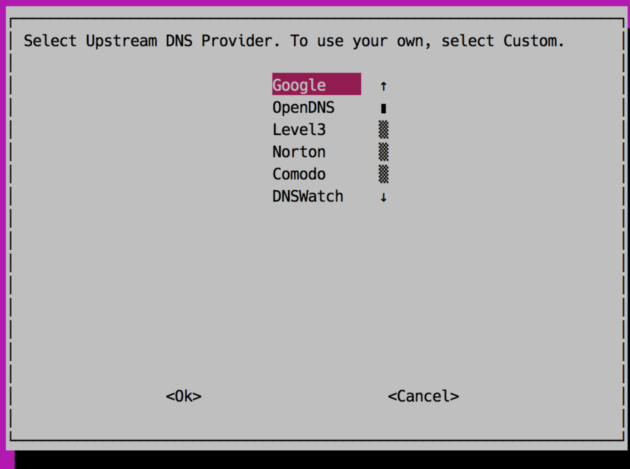
Για την εγκατάσταση του Pi-hole στο Raspberry Pi ή σε κάποια από τις υποστηριζόμενες διανομές Linux, από το επίσημο site του πρότζεκτ αρκεί να κατεβάσουμε το αντίστοιχο BASH script και να το τρέξουμε:

```
$ curl -sSL https://install.pi-hole.net --output pihole.sh
$ chmod 700 pihole.sh
$ sudo ./pihole.sh
```

Με εξαίρεση ένα-δύο σημεία που αξίζει να προσέξουμε, η διαδικασία εγκατάστασης του Pi-hole είναι απλή και δεν χρειάζεται ιδιαίτερες εξηγήσεις. Πριν την ξεκινήσετε δείτε τα screenshots που ακολουθούν, διαβάστε και τις αντίστοιχες περιγραφές.



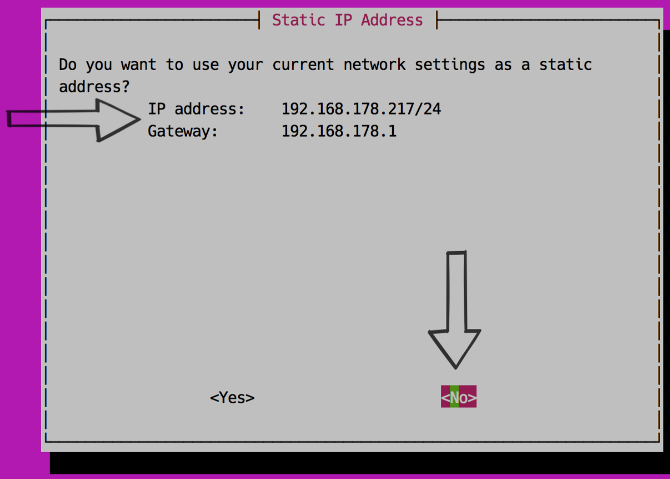
Όπως και το PiVPN (**https://deltahacker.gr/raspi-easyvpn**), έτσι και το Pi-hole ζητά έναν host με στατική διεύθυνση IP. Αν για την περίπτωσή σας πράγματι χρειάζεται να ορίσετε στατικό IP, σε λίγο θα έχετε την ευκαιρία να το κάνετε.



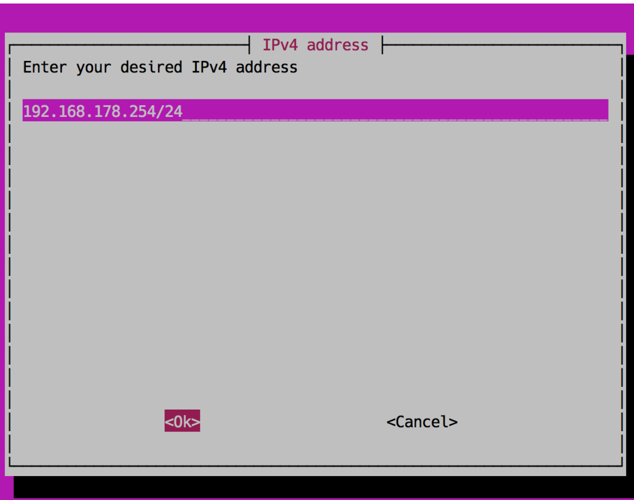
Το Pi-hole λειτουργεί ως nameserver για τις συσκευές του τοπικού δικτύου, αλλά κι αυτό χρειάζεται να συμβουλεύεται άλλους nameservers. Εδώ λοιπόν επιλέγουμε upstream nameservers για το Pi-hole, ενώ σημειώστε ότι μπορούμε να υποδείξουμε και κάποιους που δεν περιλαμβάνονται στους προτεινόμενους. Για εμάς, πάντως, οι nameservers της Google δουλεύουν μια χαρά.



Επιπλέον του κλασικού IPv4, αρκετοί ISPs προσφέρουν στους πελάτες τους και υποστήριξη για το IPv6. Κάθε σύγχρονο λειτουργικό σύστημα υποστηρίζει αμφότερα τα IP stacks, οπότε καλό είναι να ζητήσουμε από το Pi-hole να μπλοκάρει διαφημίσεις τόσο πάνω από το IPv4, όσο και πάνω από το IPv6.



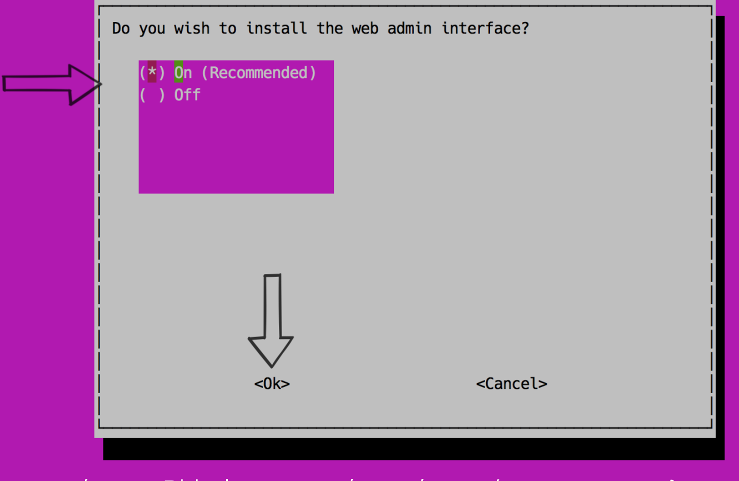
Ιδού η δυναμική διεύθυνση IP (192.168.178.217) που έχει πάρει το Raspberry Pi μας από το modem/router του τοπικού δικτύου (192.168.178.1). Θέλουμε να ορίσουμε μόνοι μας στατική διεύθυνση IP, γι' αυτό και πατάμε στο <No>.



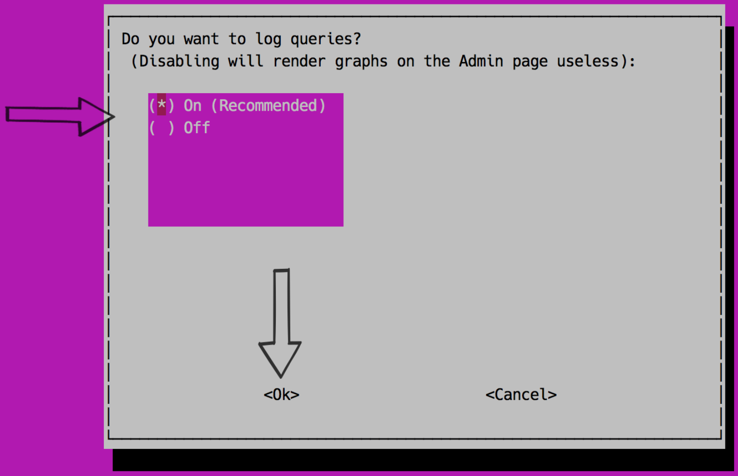
Η στατική διεύθυνση IP που ορίσαμε για το Raspberry Pi είναι η 192.168.178.254 (και βρίσκεται εκτός του εύρους διευθύνσεων από το οποίο ο DHCP του modem/router μοιράζει διευθύνσεις.)



Η διεύθυνση του gateway δεν αλλάζει και είναι εκείνη που έχει το LAN interface του modem/router.



Προαιρετικά, το Pi-hole προσφέρει ένα εύχρηστο και λειτουργικό web dashboard που αξίζει να έχουμε.



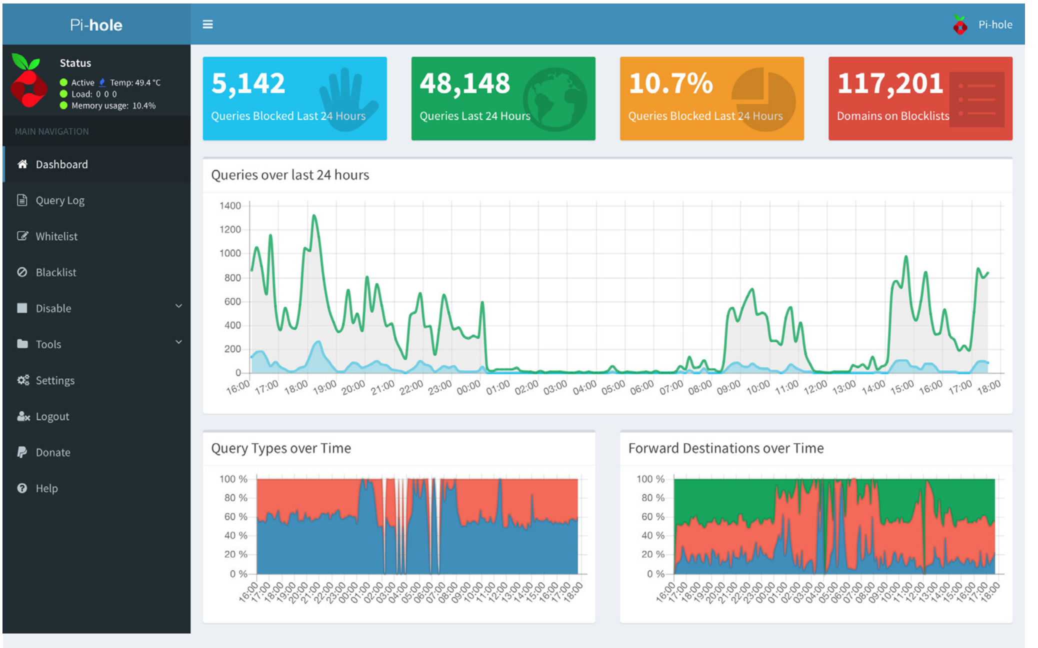
Αν μη τι άλλο για τα ωραία γραφήματα και στατιστικά που παράγει το Pi-hole, είναι πολύ χρήσιμο να καταγράφουμε τα requests προς domains που κάνουν οι συσκευές του τοπικού δικτύου.



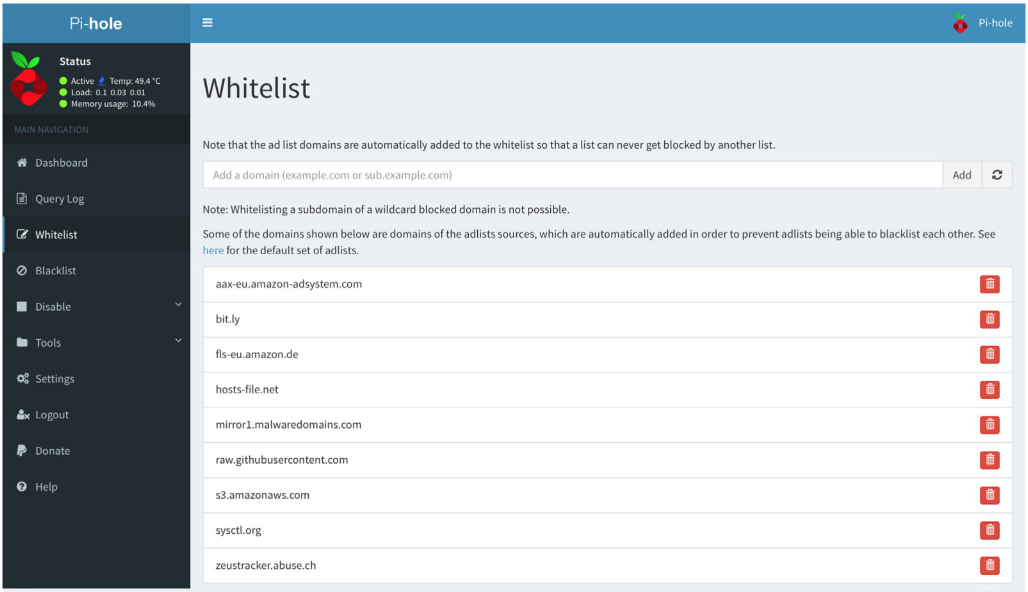
Η εγκατάσταση του Pi-hole μόλις ολοκληρώθηκε. Στην περίπτωσή μας, το σχετικό web dashboard είναι διαθέσιμο από τη διεύθυνση **http://192.168.178.254/admin**. Για την είσοδο στο dashboard απαιτείται login στο λογαριασμό χρήστη με username το admin και password αυτό που αυτόματα παρήγαγε ο installer του Pi-hole. Σημειώστε ότι ανά πάσα στιγμή μπορούμε να αλλάζουμε password: απλά συνδεόμαστε στο Raspbian μέσω SSH και στη γραμμή εντολών πληκτρολογούμε pihole -a -p.

Προκειμένου να δοκιμάσετε άμεσα τις ικανότητες ad blocking του Pi-hole, επισκεφτείτε ένα site το οποίο γνωρίζετε πως δεν τσιγκουνεύεται με το σερβίρισμα διαφημίσεων. Αμέσως μετά φροντίσετε ώστε ο υπολογιστής ή η συσκευή σας να έχει ως nameserver το Raspberry Pi (κάντε τη σχετική ρύθμιση χειροκίνητα). Φορτώστε ξανά το site και παρατηρήστε τις διαφορές.

Συνδεθείτε στο dashboard του Pi-hole (http://IP_του_pihole/admin) και παρατηρήστε τα γραφήματα και τα στατιστικά. Όσο περισσότερο χρησιμοποιείτε τον nameserver που πλέον κατοικοεδρεύει στο Raspberry Pi σας, τόσο περισσότερο ενδιαφέρον θα αποκτούν τα στατιστικά. Είναι εκπληκτικό το τι συμβαίνει με τις συνδέσεις μας προς το Internet, και μάλλον σοκαριστικό το ποσοστό του bandwidth που σπαταλιέται χωρίς λόγο και για κανένα ουσιαστικό όφελος.



Να ένα μικρό οικιακό δίκτυο, στο οποίο συνδέονται επτά συσκευές, περίπου άλλα τόσα VMs αλλά **μόλις δύο** χρήστες. Παρά το γεγονός ότι τείνουν να επισκέπτονται λίγο πολύ τα ίδια sites και να χρησιμοποιούν τις ίδιες υπηρεσίες, περισσότερο από το 10% των DNS requests αφορούν σε ad domains. Αν γίνονταν επισκέψεις σε περισσότερα sites, το ποσοστό αυτό ίσως ήταν χαμηλότερο — αν και κάτι μας λέει πως θα ήταν μεγαλύτερο. Σε κάθε περίπτωση, αυτό το 10,7% δεν είναι αμελητέο. Χάρη όμως στο Pi-hole τα σχετικά requests απορρίπτονται κι έτσι άχρηστα ή ενοχλητικά banners, εικόνες, animations και βίντεο, δεν φτάνουν ποτέ στο οικιακό δίκτυο.

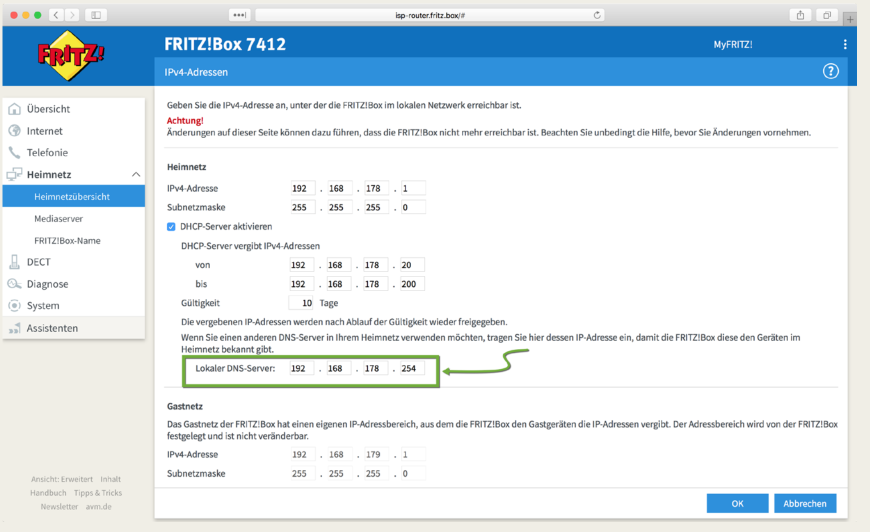


Μεταξύ των χρήσιμων λειτουργιών που παρέχονται από το dashboard του Pi-hole είναι εκείνη του whitelisting. Παρατηρήστε, για παράδειγμα, ότι όσον αφορά στο Amazon θέλουμε "όλη την εμπειρία", ακόμη κι αν μιλάμε για διαφημίσεις. Δείτε επίσης ότι σε whitelist έχουμε και την υπηρεσία URL shortening του bit.ly. Συνειδητοποιήσαμε ότι το εν λόγω domain εξ ορισμού μπλοκάρεται, όταν επιχειρήσαμε να διαβάσουμε άρθρο που είδαμε σε έγκυρο post στο Facebook και η αντίστοιχη σελίδα δεν φόρτωνε.

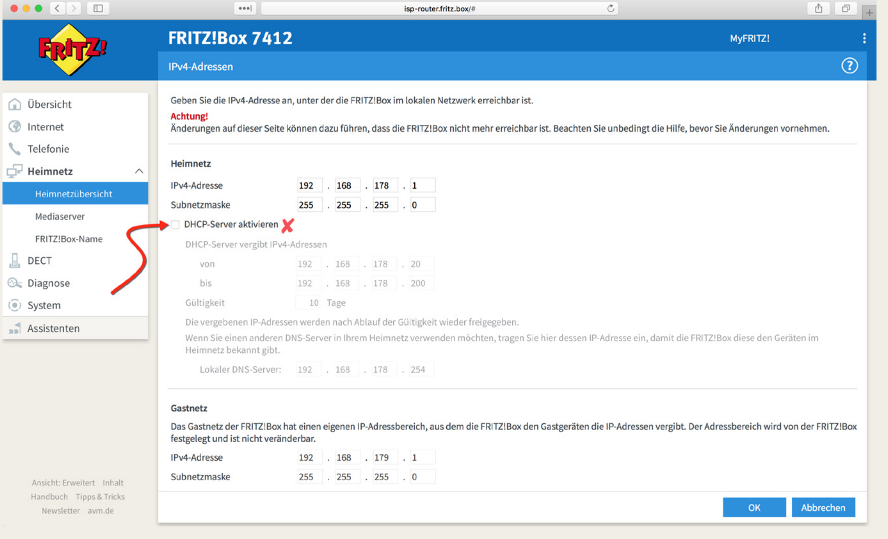
Μπλοκάρισμα διαφημίσεων για όλο το δίκτυο

Αφού δοκιμάσετε το Pi-hole μεμονωμένα για κάποιον υπολογιστή ή συσκευή, στοιχηματίζουμε ότι σύντομα θα αποφασίσετε να το αναδείξετε ως τον προκαθορισμένο nameserver για το τοπικό σας δίκτυο. Πρακτικά, δεν θα θέλετε να υποδεικνύετε **εσείς** σε κάθε υπολογιστή ή συσκευή να τον χρησιμοποιεί για τα DNS queries. Πολύ περισσότερο, θα θελήσετε να **επιβάλετε** με κάποιον τρόπο τη χρήση του, αυτόματα. Δύο μεθόδους –ή καλύτερα στρατηγικές– μπορείτε ν' ακολουθήσετε, κι αμφότερες εμπλέκουν το υπάρχον modem/router του τοπικού σας δικτύου.

Υπόδειξη nameserver από πλευράς modem/router. Όλα παραμένουν ως έχουν στο δίκτυό μας, με μόνη τη διαφορά ότι ο DHCP server του modem/router γνωστοποιεί στους πελάτες τον nameserver που βρίσκεται στο Raspberry Pi – μ' άλλα λόγια δίνει το IP του Raspbian. Βεβαίως η ρύθμιση αυτή πραγματοποιείται διαφορετικά από συσκευή σε συσκευή, όμως η λογική παραμένει ίδια: αντί οι DHCP clients να βλέπουν το modem/router ως nameserver, θέλουμε βλέπουν το Raspberry Pi.

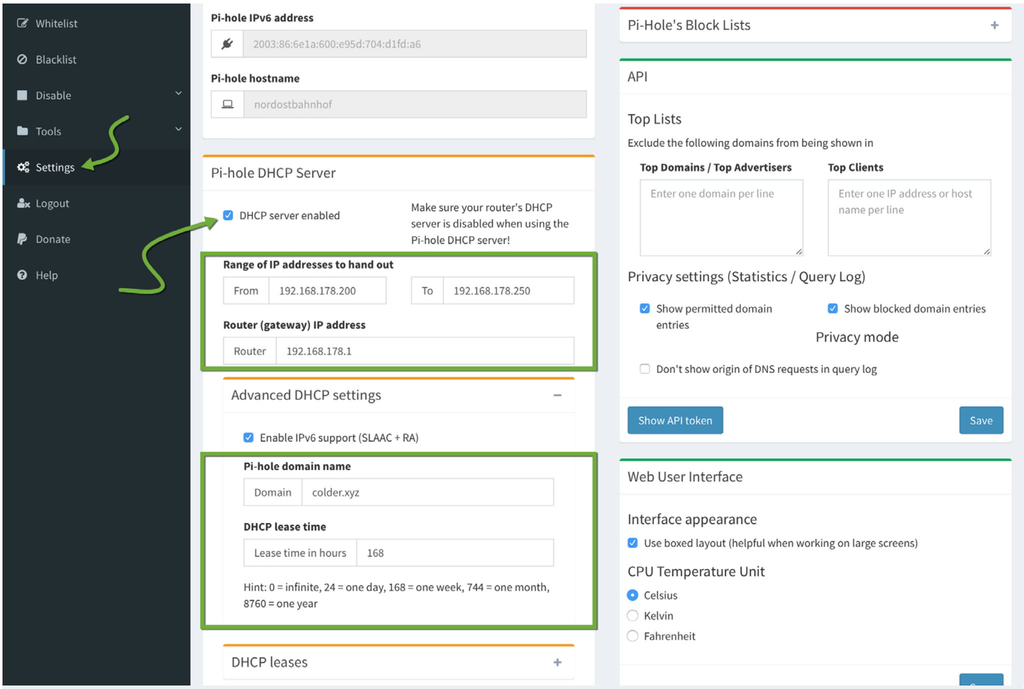


Μετά από αρκετό Google translate βγάλαμε άκρη και καταφέραμε να πούμε στο modem/router μας να δίνει το 192.168.178.254 ως nameserver στους DHCP clients. Στη διεύθυνση αυτή απαντάει ο nameserver του Pi-hole, οπότε όλες οι συσκευές στο τοπικό μας δίκτυο χρησιμοποιούν πλέον αυτόν.



Απενεργοποίηση του DHCP server στο modem/router που έχουμε στο τοπικό μας δίκτυο. Φυσικά, η συσκευή συνεχίζει να λειτουργεί ως gateway. Σε λίγο εξάλλου θα έχουμε και πάλι DHCP server, μόνο που τώρα αυτόν το ρόλο θα τον παίζει το Pi-hole.

Υποβάθμιση ρόλου modem/router, στροφή στον DHCP του Pi-hole. Αντί να πούμε στο modem/router να διαφημίζει το Pi-hole ως nameserver, μπορούμε απλά να απενεργοποιήσουμε τον DHCP server της συσκευής και να ενεργοποιήσουμε τον DHCP server του Pi-hole. Το modem/router θα συνεχίσει να λειτουργεί ως gateway ή αλλιώς ως router για το τοπικό δίκτυο, αλλά έως εκεί: με τη εξυπηρέτηση των DHCP clients θα ασχολείται πλέον το Pi-hole! Κατ' αυτόν τον τρόπο αφενός αποκτάμε έναν πιο ευέλικτο DHCP server (μπορούμε, π.χ., να ορίσουμε το domain της επιλογής μας για τους clients ή/και να δημιουργούμε static IP mappings), αφετέρου οι συσκευές του τοπικού δικτύου απευθύνουν τα DNS queries στο Pi-hole και, συνεπώς, οι χρήστες τους λένε αντίο στις διαφημίσεις.



Ενεργοποίηση του DHCP server (dnsmasq) που διαθέτει το Pi-hole. Παρατηρήστε το εύρος διευθύνσεων από το οποίο ο DHCP εκλέγει και μοιράζει IPs στους πελάτες του: αποτελεί υποσύνολο του δικτύου που ορίζει το LAN interface του modem/router. Για τους DHCP clients, εξάλλου, υποδείξαμε τη διεύθυνση του router (παραμένει η συσκευή του ISP μας), ένα domain (colder.xyz), καθώς και το lease time.

Συνεργασία PiVPN και Pi-hole

Ο συνδυασμός PiVPN και Pi-hole δεν ενισχύει μόνο την ιδιωτικότητά μας, όταν βρισκόμαστε εκτός home network. Συμβάλλει και στην καλύτερη αξιοποίηση του διαθέσιμου bandwidth για συσκευές μας που συνδέονται στο Internet, άμεσα ή έμμεσα, μέσω του δικτύου κάποιας εταιρείας κινητής τηλεφωνίας. Δεν θα διστάζαμε μάλιστα να προσθέσουμε ότι όφελος παρατηρείται και στη διάρκεια ζωής της μπαταρίας, αφού ο browser δεν ασχολείται με το rendering ή/και το playback αδιάφορου περιεχομένου.

Υποθέτουμε λοιπόν ότι μαζί με το Pi-hole έχετε εγκαταστήσει **και** το PiVPN (https://deltahacker.gr/raspi-easyvpn). Υπόψιν ότι τα δύο εργαλεία μπορούν και συνυπάρχουν άνετα στο ίδιο σύστημα, χωρίς διενέξεις. Εξ ορισμού όμως το PiVPN **δεν** χρησιμοποιεί το Pi-hole — και μην ξεγελιέστε από τις εξωτερικές ομοιότητες των scripts εγκατάστασης. Προκειμένου ν' αλλάξουμε αυτή την κατάσταση, ώστε να συνδεόμαστε στο PiVPN κι αυτομάτως να χρησιμοποιούμε το Pi-hole ως nameserver, συνδεόμαστε στο Raspberry Pi μέσω SSH και φροντίζουμε για τα ακόλουθα.

- Με δικαιώματα root ανοίγουμε το αρχείο `/etc/dnsmasq.conf`. Εντοπίζουμε τη γραμμή που ξεκινά με `#listen-address=` και την αλλάζουμε ώστε να μοιάζει με αυτή: `listen-address=127.0.0.1, 10.8.0.1, 192.168.178.254`. Εσείς, βεβαίως, στη θέση του `192.168.178.254` θα βάλετε τη διεύθυνση IP που έχει το Raspberry Pi σας. Αποθηκεύουμε την αλλαγή κι επανεκκινούμε το dnsmaq με ένα `sudo systemctl restart dnsmasq`.
- Ξανά με δικαιώματα root ανοίγουμε το αρχείο `/etc/openvpn/server.conf` και, αναφορικά με τον DNS server που γνωστοποιείται στους OpenVPN clients, φροντίζουμε ώστε να είναι ο `10.8.0.1`. Μ' άλλα λόγια, στο αρχείο πρέπει να υπάρχει αυτή η γραμμή: `push "dhcp-option DNS 10.8.0.1"` (αν το IP είναι διαφορετικό, αντικαταστήστε το με το `10.8.0.1`). Αφού αποθηκεύσουμε την αλλαγή εγκαταλείπουμε τον editor κι επανεκκινούμε τον OpenVPN server πληκτρολογώντας `sudo systemctl restart openvpn`.

Όλα έτοιμα, μπορούμε πλέον να απολαμβάνουμε ενισχυμένη ιδιωτικότητα, web χωρίς διαφημίσεις και φυσικά more bandwidth for our buck (χαρήκαμε που σας γνωρίσαμε, αντίο σας).

Skill: Beginner
Tags: GRUB2, openSUSE, Ubuntu

Αποτροπή root password reset σε συστήματα Linux

Σε προηγούμενο άρθρο δείξαμε πώς βάζουμε νέο password για το χρήστη root ενός συστήματος Linux, όταν δεν θυμόμαστε το τρέχον. Εν τάχει, ξεκινάμε τροποποιώντας κατάλληλα το σχετικό boot entry στο μενού του GRUB2 και δευτερόλεπτα αργότερα ορίζουμε νέο password για τον root. Η τεχνική είναι ιδιαίτερα χρήσιμη για τους ξεχασιάρηδες, σημαίνει όμως ότι οποιοσδήποτε έχει φυσική πρόσβαση στον υπολογιστή είναι σε θέση να αποκτήσει πρόσβαση στο λογαριασμό του διαχειριστή.

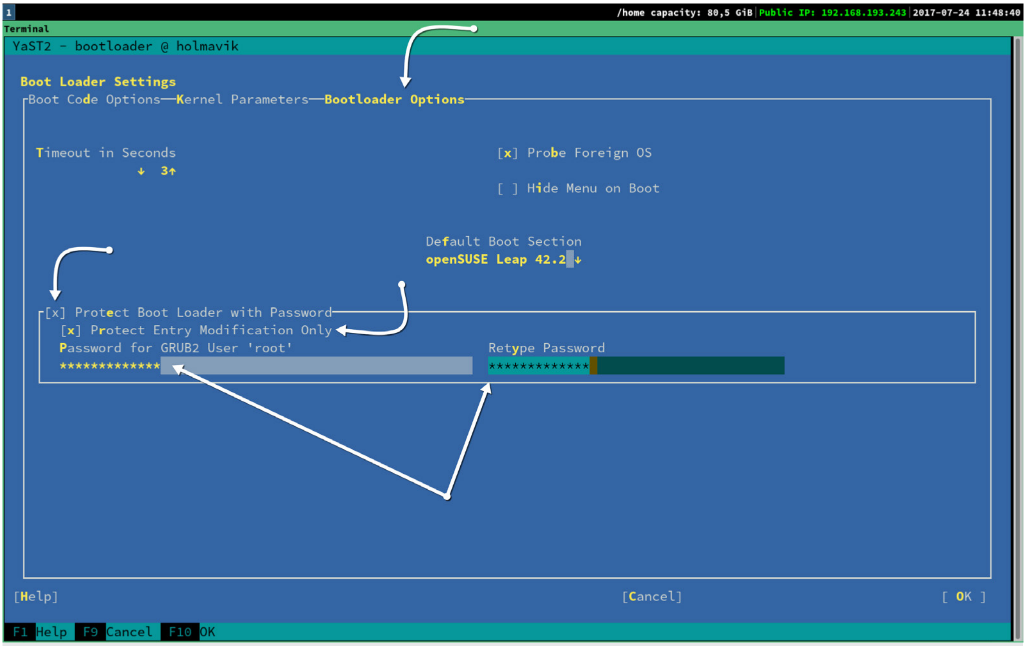
Όπως κι εσείς μπορείτε να διαπιστώσετε (https://deltahacker.gr/?p=17533), αρκεί κάποιος να καθίσει μπροστά από ένα σύστημα Linux και θα καταφέρει να ορίσει νέο password για τον `root` **χωρίς** να γνωρίζει το παλιό. Ως πρώτη σκέψη για αντίμετρο θα μπορούσαμε να σκεφτούμε password για το BIOS (ή για το UEFI) του PC. Πράγματι, ξεκινώντας ή επανεκκινώντας ένα PC με password στο BIOS, καλούμαστε να το πληκτρολογήσουμε **πριν** φτάσουμε στην οθόνη του GRUB2 ή όποιου άλλου boot manager. Το θέμα είναι ότι τα BIOS passwords είναι σχετικά εύκολο να διαγραφτούν, ειδικά αν μιλάμε για PC με ανεμπόδιστη πρόσβαση στο motherboard. Αναλυτικότερα, το password ακυρώνεται με επαναφορά των εργοστασιακών ρυθμίσεων του BIOS. Η δε επαναφορά επιτυγχάνεται αφαιρώντας προσωρινά τη μπαταρία του motherboard ή αλλάζοντας θέση στο κατάλληλο jumper. Σίγουρα ένας κακόβουλος χρήστης επιβραδύνεται σημαντικά όταν χρειάζεται να κάνει reset στο BIOS, αλλά όπως και να 'χει η συγκεκριμένη μέθοδος προστασίας είναι ανεπαρκής.

Μια εναλλακτική λύση, η οποία μπορεί να μην είναι τέλεια αλλά αλλά ανεβάζει το επίπεδο δυσκολίας για attackers χωρίς επαρκείς γνώσεις περί Linux, αφορά στον καθορισμό password για το GRUB2. Συγκεκριμένα, αναφερόμαστε σε password το οποίο θα ζητείται κατά την ενεργοποίηση boot entry ή/και κατά την απόπειρα τροποποίησής του. Δείτε στη συνέχεια πώς ορίζουμε password για το GRUB2 όταν η διανομή μας είναι το **openSUSE** ή το **Ubuntu**.

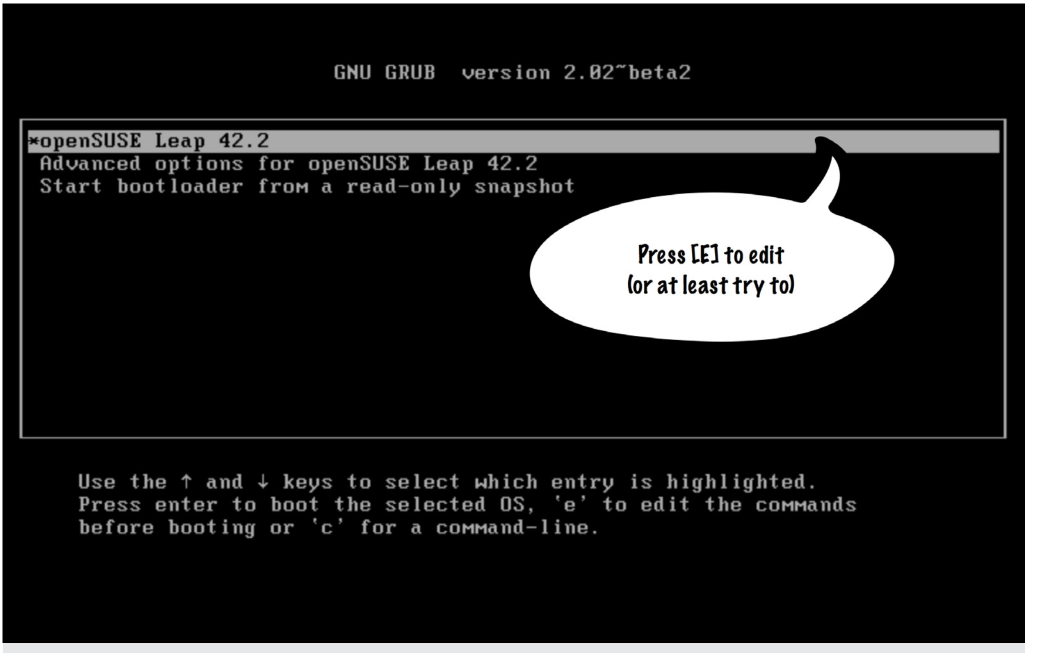
Περίπτωση openSUSE

Τα πράγματα δεν θα μπορούσαν να είναι απλούστερα. Για την προστασία των boot entries του GRUB2 από τροποποιήσεις, αρκεί να ενεργοποιήσουμε το module ονό-

ματι `bootloader` του YaST2: σε ένα τερματικό απλά πληκτρολογούμε `sudo yast2 bootloader`. Δείτε τα τρία screenshots που ακολουθούν, διαβάστε και τις αντίστοιχες περιγραφές.



Στο module bootloader του YaST2 πηγαίνουμε κατευθείαν στην κατηγορία Bootloader Options (απλά πατάμε το συνδυασμό πλήκτρων [ALT+L]). Θέλουμε βεβαίως να ενεργοποιήσουμε αυτό το Protect Boot Loader with Password, οπότε πατάμε [ALT+E]. Επειδή μας ενδιαφέρει η δυνατότητα για απομακρυσμένα reboot, ενεργοποιούμε το Protect Entry Modification Only ([ALT+R]). Υπάρχει ένας χρήστης του GRUB2 ο οποίος ονομάζεται "root" και γι' αυτόν ορίζουμε ένα password πατώντας [ALT+P]. Φυσικά, για λόγους ασφαλείας καλούμαστε να πληκτρολογήσουμε ξανά το ίδιο ακριβώς password ([ALT+Y]). Σημειώστε ότι κατά την πληκτρολόγηση του password, στη θέση των χαρακτήρων εμφανίζονται αστερίσκοι. Επικυρώνουμε τις επιλογές μας με το συνδυασμό [ALT+O].



Αντικρίζουμε το ωραίο, text-based μενού του GRUB2. Θέλουμε να τροποποιήσουμε το πρώτο boot entry, με στόχο ν' αλλάξουμε το password του χρήστη root (https://deltahacker.gr/?p=17533). Το επιλέγουμε, λοιπόν, κι αμέσως πατάμε το πλήκτρο [E] (από το "Edit").



Η απόπειρά μας για τροποποίηση του boot entry προκαλεί τον GRUB2 να μας ζητήσει username και password. Για το πρώτο δίνουμε "root" (χωρίς εισαγωγικά και μιλάμε για τον root του GRUB2, όχι για τον ομώνυμο χρήστη του συστήματος), ενώ για password πληκτρολογούμε αυτό που ορίσαμε πριν από λίγο, στο module ονόματι `bootloader` του YaST2. Μόνον εφόσον πληκτρολογήσουμε σωστά username και password θα μας επιτραπεί να κάνουμε αλλαγές στο boot entry.

Περίπτωση Ubuntu

Στο πλαίσιο των δοκιμών μας εργαστήκαμε σε ένα σύστημα με Zesty Zapus, δηλαδή με Ubuntu 17.04. Παρόμοια θα εργαστεί κάποιος σε άλλες εκδόσεις του Ubuntu – ακόμη και σε άλλες διανομές του Linux που χρησιμοποιούν το GRUB2. Ξεκινάμε τροποποιώντας ελαφρά το αρχείο /etc/default/grub, με δικαιώματα διαχειριστή φυσικά: sudo vi /etc/default/grub. Φροντίζουμε ώστε η γραμμή...

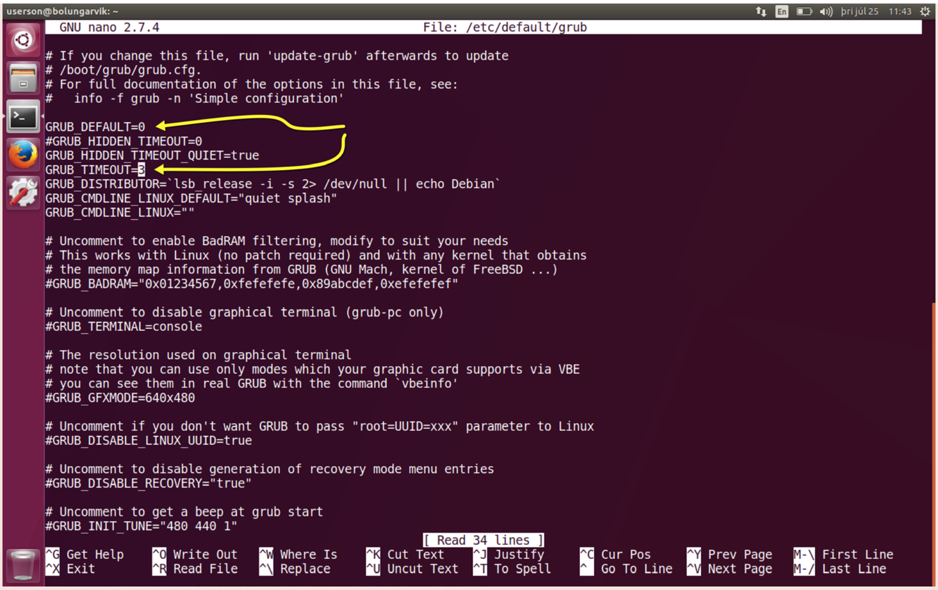
```
GRUB_HIDDEN_TIMEOUT=0
```

...πάνω πάνω, να γίνει έτσι:

```
#GRUB_HIDDEN_TIMEOUT=0
```

Σχολιάζοντάς τη, λοιπόν, έχουμε την ευκαιρία να βλέπουμε το μενού του GRUB2 κάθε φορά που ανοίγουμε ή επανεκκινούμε τον υπολογιστή. Προαιρετικά, αλλάζουμε και το χρόνο παραμονής του μενού πριν ξεκινήσει το φόρτωμα του προκαθορισμένου boot entry. Εμείς, π.χ., από τα 10 δευτερόλεπτα τον κατεβάσαμε στα 3, οπότε στο /etc/default/grub έχουμε την ακόλουθη γραμμή:

```
GRUB_TIMEOUT=3
```

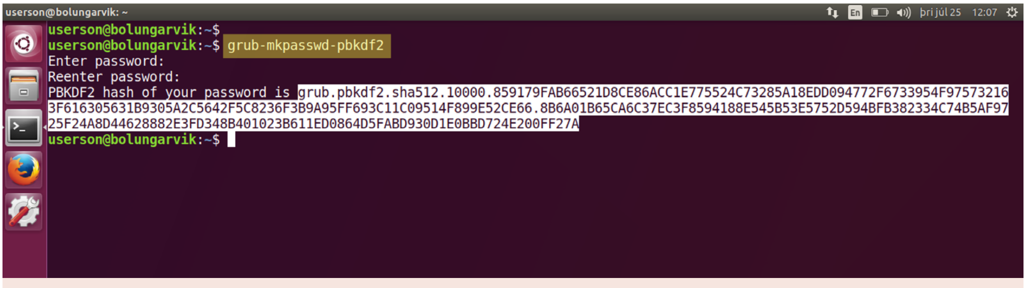


Θέλουμε να βλέπουμε το μενού του GRUB2 στην εγκατάσταση του Ubuntu μας, αλλά μετά τρία δευτερόλεπτα αδράνειας να φορτώνει το προεπιλεγμένο boot entry.

Αποθηκεύουμε τις αλλαγές στο `/etc/default/grub` και χωρίς καμία καθυστέρηση παράγουμε ένα PBKDF2 (https://en.wikipedia.org/wiki/PBKDF2) hash για το password που θέλουμε να έχει ο διαχειριστής του GRUB2. Δεν χρειάζεται ν' αποφασίσουμε από τώρα για το username του εν λόγω διαχειριστή, χρειαζόμαστε όμως ένα PBKDF2 hash για το password του. Προς τούτο, στο τερματικό μας πληκτρολογούμε:

```
grub-mkpasswd-pbkdf2
```

Το εργαλείο θα μας ζητήσει να πληκτρολογήσουμε το επιθυμητό password και μετά άλλη μια φορά, για λόγους ασφαλείας (καθώς πληκτρολογούμε, οι χαρακτήρες του password δεν εμφανίζονται στην οθόνη). Με την προϋπόθεση ότι δεν θα κάνουμε κάποιο λάθος, θα πάρουμε ένα μήνυμα της μορφής `PBKDF2 hash of your password is grub.pbkdf2.sha512.10000.the_rest_of_it`.

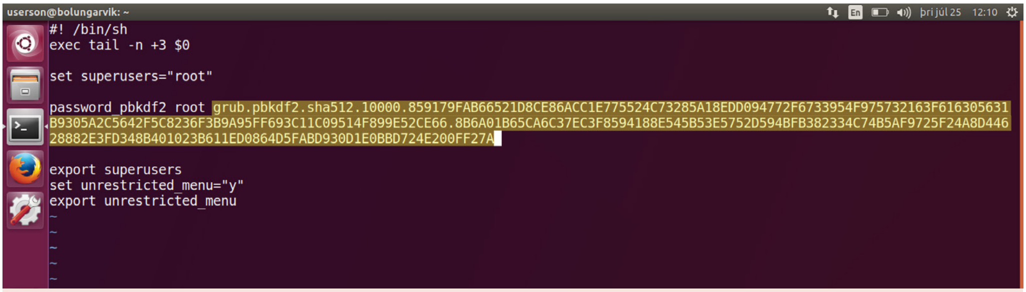


Μόλις δημιουργήσαμε ένα PBKDF2 hash, ξεκινώντας από το password που πληκτρολογήσαμε για το χρήστη με δικαιώματα διαχείρισης του GRUB2. Παρατηρήστε την τονισμένη περιοχή: αυτό ακριβώς είναι το PBKDF2 hash!

Στη συνέχεια πληκτρολογούμε `sudo vi /etc/grub.d/42_password`, ώστε με δικαιώματα `root` να δημιουργήσουμε το αρχείο `42_password` εντός του καταλόγου `/etc/grub.d`. Φροντίζουμε ώστε το περιεχόμενό του να μοιάζει με το ακόλουθο:

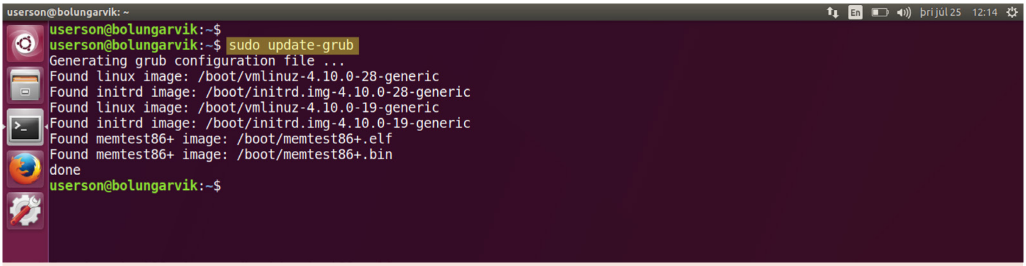
```
#! /bin/sh
exec tail -n +3 $0
set superusers="root"
password_pbkdf2 root grub.pbkdf2.sha512.10000.the_rest_of_it
export superusers
set unrestricted_menu="y"
export unrestricted_menu
```

Προσοχή ώστε στη θέση του `grub.pbkdf2.sha512.10000.the_rest_of_it` να βάλετε ολόκληρο το string που σας επέστρεψε το `grub-mkpasswd-pbkdf2` (δείτε ξανά με προσοχή το προηγούμενο screenshot, διαβάστε και το συνοδευτικό κείμενο). Παρατηρήστε εξάλλου ότι τον έναν και μοναδικό διαχειριστή του GRUB2 (βλ. οδηγία `set superusers`) ονομάσαμε `root`. Εσείς φυσικά μπορείτε να του δώσετε κάποιο άλλο username – και μην παραλείψετε να βάλετε το σωστό username και στη γραμμή από κάτω, στα δεξιά του `password_pbkdf2` και στ' αριστερά του `grub.pbkdf2.sha512.10000`.



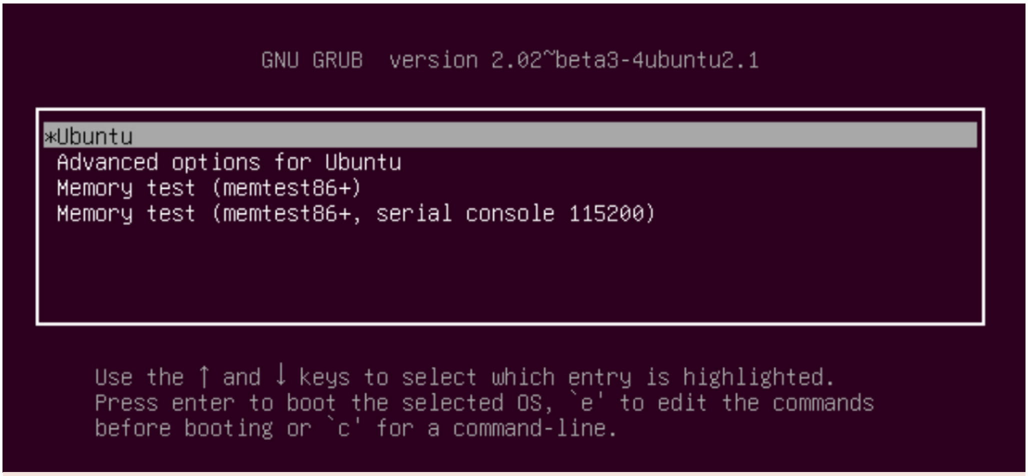
Στο αρχείο /etc/grub.d/42_password ορίζεται το username του χρήστη με δικαιώματα διαχείρισης του GRUB2, ενώ παρατίθεται και το PBKDF2 hash που παρήχθη με το εργαλείο grub-mkpasswd-pbkdf2 ξεκινώντας από το password του προαναφερθέντος χρήστη.

Αποθηκεύουμε τις αλλαγές στο /etc/grub.d/42_password κι εγκαταλείπουμε τον editor. Φροντίζουμε και για τα σωστά δικαιώματα του νέου αρχείου, δίνοντας sudo chmod 700 /etc/grub.d/42_password. Οι αλλαγές που μόλις κάναμε δεν έχουν ακόμη ληφθεί υπόψη από τον GRUB2, οπότε τον ενημερώνουμε σχετικά πληκτρολογώντας sudo update-grub.



Ενημέρωση του GRUB2 για τις αλλαγές που μόλις πραγματοποιήσαμε.

Πάμε τώρα να ελέγξουμε αν όλα δουλεύουν κατά το αναμενόμενο. Επανεκκινούμε το σύστημα και, λογικά, θα δούμε το μενού του GRUB2.



Ιδού το μενού του GRUB2 στην εγκατάσταση του Ubuntu που χρησιμοποιήσαμε για τις δοκιμές μας.

Πατώντας το πλήκτρο [Enter] (ενεργοποίηση) ή το [E] (τροποποίηση) πάνω από ένα οποιοδήποτε boot entry, ο GRUB2 ζητά να πληκτρολογήσουμε σωστά το username κάποιου διαχειριστή (root, για το παράδειγμά μας) καθώς και το αντίστοιχο password. Το ίδιο συμβαίνει και μετά το πέρας του προκαθορισμένου χρόνου παραμονής του μενού (3 δευτερόλεπτα).



Πατώντας το πλήκτρο Enter (ενεργοποίηση) ή το E (τροποποίηση) πάνω από ένα οποιοδήποτε boot entry, ο GRUB2 ζητά να πληκτρολογήσουμε σωστά το username κάποιου διαχειριστή καθώς και το αντίστοιχο password. Το ίδιο συμβαίνει και μετά το πέρας του προκαθορισμένου χρόνου παραμονής του μενού.

Και μια παρατήρηση για το τέλος: Ένας αποφασισμένος attacker με επαρκείς γνώσεις περί Linux, φυσική πρόσβαση στον υπολογιστή μας και με λίγο χρόνο στη διάθεσή του, θα καταφέρει να ακυρώσει **και** αυτό το μέτρο προστασίας (θα bootάρει από το Live DVD κάποιας άλλης διανομής και θα προβεί σε συγκεκριμένες ενέργειες, οι οποίες είναι εκτός θέματος του παρόντος άρθρου). Αν θέλουμε απόλυτη προστασία για μια εγκατάσταση Linux, τότε ο καλύτερος σύμμαχός μας είναι η κρυπτογράφηση του δίσκου ή των δίσκων του συστήματος (https://deltahacker.gr/?p=16940). Βέβαια με την κρυπτογράφηση δίσκων, κάθε φορά που ξεκινάμε τον υπολογιστή οφείλουμε να πληκτρολογούμε το κατάλληλο password για την αποκρυπτογράφηση. Αλλά έτσι είναι: περισσότερη ασφάλεια σημαίνει λιγότερη ευκολία – και φυσικά ισχύει και το αντίστροφο.

Skill: Intermediate
Tags: dnsmasq, lighttpd, PXE, PXE server, Raspberry Pi

Υπηρεσίες PXE server, από το Raspberry Pi

Το Raspberry Pi μας είναι υπερήφανο για τις υπηρεσίες ad-blocking και VPN που απλόχερα προσφέρει. Την ίδια στιγμή είναι πολύ άνετο σε system resources και μερικές φορές, θα το πούμε κι ας κριθούμε, απορούμε αν αισθάνεται τύψεις. Υπάρχουν βλέπετε ένα σωρό άλλες υπηρεσίες που θα μπορούσε να παρέχει – και μάλιστα χωρίς να ζορίζεται ιδιαίτερα.

Τις προάλλες το πήραμε απόφαση: Αντί το Raspberry Pi να κάθεται κι εμείς να απορούμε, πολύ περισσότερο ενδιαφέρον θα ήταν να του βρούμε ακόμη έναν ρόλο. Θέλαμε, φυσικά, να εξακολουθήσει να μπλοκάρει διαφημίσεις (https://deltahacker.gr/?p=17672) για **όλο** το home network, βεβαίως και να συνεχίσει να είναι ο έμπιστος OpenVPN server (https://deltahacker.gr/?p=17607) μας για όποτε είμαστε εκτός οικιακού δικτύου. Αρχικά πέρασαν από το μυαλό ιδέες του στιλ FTP/SAMBA/NFS server, media tank με Plex κ.ο.κ. Όμως με αυτά και με παρεμφερή θέματα έχουμε ασχοληθεί ξανά στο παρελθόν. Τώρα ψάχναμε κάτι εντελώς διαφορετικό, ώστε να μάθουμε και κάτι καινούργιο. Μετά από αρκετή σκέψη και μερικές δοκιμές, αποφασίσαμε για τον πρόσθετο ρόλο που θέλαμε ν' αναλάβει το Raspberry Pi μας. Να σας πούμε ότι, τώρα που διαβάζετε τις γραμμές αυτές, τον διεκπεραιώνει με απόλυτη επιτυχία – κι εμείς είμαστε βέβαιοι ότι καθόλου δεν αισθάνεται τύψεις.

Το Raspberry Pi στο τοπικό μας δίκτυο, φίλες και φίλοι, είναι πλέον και **PXE server**. Σε λίγο θα σας δείξουμε πώς να χαρίσετε κι εσείς ανάλογες δυνατότητες στο δικό σας RasPi – ή σε κάποιο άλλο box με Linux που έχετε στη διάθεσή σας, τέλος πάντων. Πρώτα όμως πρέπει να πούμε γι' αυτό το PXE και γιατί κάποιος θα το ήθελε.

Ο όρος PXE (προφέρεται "πίξι") είναι αρκτικόλεξο του **Preboot eXecution Environment** κι αποτελεί μια τυποποιημένη προδιαγραφή για την εκτέλεση λογισμικού που βρίσκεται αποθηκευμένο σε έναν PXE server, από έναν PXE client. Ίσως αυτό δεν φαντάζει και πολύ εντυπωσιακό. Δεν χρειάζεται άλλωστε να σκεφτούμε πολύ, ώστε να συνειδητοποιήσουμε ότι, ουσιαστικά, σχεδόν κάθε μέρα χρησιμοποιούμε λογισμικό που δεν υπάρχει στον υπολογιστή μας αλλά σε κάποιον άλλον, στο τοπικό δίκτυο ή στο Internet. Το PXE όμως δεν περιγράφει πώς τρέχουμε λογισμικό που βρίσκεται σε άλλο μηχάνημα – κι ενδεχομένως βλέπουμε τα αποτελέσματα της εκτέλεσης στην οθόνη κάποιου άλλου. Πολύ περισσότερο, περιγράφει πώς ένας client, ο οποίος **ίσως δεν έχει καν λειτουργικό σύστημα** εγκατεστημένο, ξεκινά και τρέχει κώδικα από κάποιον απομακρυσμένο host. Το μόνο που χρειάζεται ο client είναι μια ασύρματη ή ενσύρματη κάρτα δικτύου που να ξέρει τι θα πει PXE. Μια τέτοια

κάρτα γνωρίζει και το πρωτόκολλο DHCP, αφού χρειάζεται έναν τρόπο ώστε να δέχεται παραμέτρους δικτύωσης. Γνωρίζει επίσης και το απλό αλλά πολύ χρήσιμο πρωτόκολλο TFTP (**Trivial File Transfer Protocol**), χάρη στο οποίο επιτυγχάνεται η μεταφορά εκτελέσιμου κώδικα από τον PXE server στον PXE client.

Δύο είναι οι βασικές εφαρμογές που βρίσκει το PXE.

* **Εκκίνηση diskless workstations**, δηλαδή λιτών σε hardware υπολογιστών που ενδεχομένως δεν έχουν καν δίσκο. Τέτοια μηχανήματα φορτώνουν το λειτουργικό τους σύστημα από έναν PXE server, ενώ είναι και πιθανό να προσαρτούν τα home directories των χρηστών από κάποιον άλλον υπολογιστή, π.χ., μέσω NFS.

* **Δικτυακή εγκατάσταση λειτουργικού συστήματος**, χωρίς να απαιτούνται το φόρτωμα και η εκκίνηση του installer από CD/DVD/USB stick ή από φορητό δίσκο. Σε πολλά data centers αυτός ακριβώς είναι ο ενδεδειγμένος τρόπος για την εγκατάσταση λειτουργικών σε υπολογιστές. Δεν χρειάζεται, δηλαδή, να τρέχει ένας διαχειριστής εδώ κι εκεί, με δισκάκια και στικάκια και να φροντίζει για τις διαδικασίες εγκατάστασης. Μάλιστα ακόμη και οι (επαν)εκκινήσεις των μηχανημάτων γίνονται απομακρυσμένα, π.χ., χάρη στο IPMI (Intelligent Platform Management Interface).

Στο παρόν άρθρο μάς ενδιαφέρει αυτή η δεύτερη εφαρμογή, κι ας μην είμαστε υπεύθυνοι διαχειριστές σε κάποιο data center. Πράγματι, ένας μικρός PXE server αποδεικνύεται χρήσιμος ακόμη και σε ένα home network, αν μη τι άλλο για να κάνουμε τις εγκαταστάσεις μας και τις δοκιμές μας δικτυακά, χωρίς να σκοτιζόμαστε με δισκάκια και στικάκια. Εννοείται ότι οι PXE clients επιτρέπεται να είναι **και** εικονικές μηχανές. Να πούμε επίσης ότι, πέρα από τις εγκαταστάσεις, χάρη στον PXE server είναι πολύ εύκολο να ξεκινάμε μια Live διανομή Linux για την επιδιόρθωση ή το backup ενός υπάρχοντος συστήματος (σκεφτείτε, π.χ., το SystemRescueCd, http://www.system-rescue-cd.org).

Υποψιαζόμαστε ότι κάπου εδώ έχουμε κερδίσει την προσοχή σας, οπότε ας περάσουμε σιγά σιγά στη δράση.

Προαπαιτούμενα

Δουλεύουμε σε ένα Raspberry Pi 3 με Raspbian, το οποίο εκτελεί χρέη network-wide ad-blocker με το Pi-hole (https://deltahacker.gr/?p=17672), καθώς κι OpenVPN server με το PiVPN (https://deltahacker.gr/?p=17607). Στο RasPi λοιπόν υπάρχει ήδη το dnsmasq εγκατεστημένο, και λειτουργεί ως DNS server για τα μηχανήματα και τις συσκευές του τοπικού δικτύου. Έχουμε επίσης ζητήσει από το Pi-hole να προσφέρει υπηρεσίες DHCP, γεγονός που στην πράξη σημαίνει ότι το dnsmasq είναι **ο ένας και μοναδικός** DHCP server του τοπικού δικτύου. (Ναι, στο modem/router του ISP φροντίσαμε για την απενεργοποίηση της αντίστοιχης δυνατότητας.) Μπορείτε να εφαρμόσετε όλα όσα ακολουθούν στο Raspberry Pi σας — ή και σε κάποιο άλλο μηχάνημα με Linux. Να σημειώσουμε εδώ ότι, αν και το dnsmasq **δεν** είναι υποχρεωτικό και **σίγουρα** υπάρχει δυνατότητα επιλογής άλλου DHCP server, οι σχετικές ρυθμίσεις που σε λίγο θα δείξουμε έχουν νόημα για το dnsmasq και μόνο γι' αυτό. Αν δεν έχετε ακόμη αποφασίσει για τον DHCP server που θα χρησιμοποιήσετε, σας προτείνουμε να εξετάσετε σοβαρά την περίπτωση του dnsmasq: πρόκειται για έναν server που

εκτός από DHCP παρέχει και υπηρεσίες DNS, ενώ ταυτόχρονα υποστηρίζει τα PXE και TFTP. Όσον αφορά τέλος στο hardware, και συγκεκριμένα στις κάρτες δικτύωσης, δεν υπάρχει κανένας λόγος ανησυχίας: είναι **πρακτικά αδύνατο** να διαθέτετε κάρτα που δεν υποστηρίζει το PXE, ενώ το ίδιο ισχύει και για τις κάρτες δικτύωσης των εικονικών μηχανών.

Απαραίτητο λογισμικό και πρώτη δικτυακή εκκίνηση

Συνδεόμαστε στο Raspberry Pi μέσω SSH, φρεσκάρουμε τα repositories, εφαρμόζουμε τις όποιες διαθέσιμες ενημερώσεις κι εγκαθιστούμε το απαραίτητο λογισμικό για τη λειτουργικότητα PXE:

```
pi@nordostbahnhof:~ $ sudo apt update
pi@nordostbahnhof:~ $ sudo apt upgrade
pi@nordostbahnhof:~ $ sudo apt install pxelinux syslinux-common
```

Με δικαιώματα διαχειριστή και με τον text editor της προτίμησής μας, ανοίγουμε το αρχείο `/etc/dnsmasq.conf`. Στο τέλος του προσθέτουμε τέσσερις νέες γραμμές που μοιάζουν με τις ακόλουθες:

```
enable-tftp
dhcp-boot=pxelinux.0
pxe-service=x86PC,"Nordostbahnhof Boot System",pxelinux
tftp-root=/mnt/external/tftpboot
```

Ως μήνυμα στην τρίτη γραμμή, ανάμεσα στα εισαγωγικά, βάλτε ό,τι θέλετε. Στην τέταρτη γραμμή καθορίστε την πλήρη διαδρομή προς τον κατάλογο κάτω από τον οποίο σκοπεύετε να αποθηκεύετε κάποια κρίσιμα αρχεία για διάφορα λειτουργικά που επιθυμείτε να προσφέρει ο PXE server. Περιττό ν' αναφέρουμε ότι καλό είναι να αποφύγετε την κάρτα SD του Raspberry Pi. Εμείς, π.χ., συνδεδεμένο στη συσκευή έχουμε έναν 500άρη εξωτερικό δίσκο USB. Τη μία και μοναδική κατάτμησή του την έχουμε προσαρτημένη κάτω από τον κατάλογο `/mnt/external`. Μεταξύ των καταλόγων στη ρίζα του δίσκου υπάρχει κι ο tftpboot — και κάτω απ' αυτόν οργανώνουμε όλους τους καταλόγους και τα αρχεία που χρησιμοποιεί ο PXE server. Αποθηκεύουμε τις αλλαγές στο dnsmasq.conf, εγκαταλείπουμε τον editor κι επανεκκινούμε την υπηρεσία:

```
pi@nordostbahnhof:~ $ sudo systemctl restart dnsmasq
```

Αλλάζουμε το ιδιοκτησιακό καθεστώς από τον κατάλογο `/mnt/external/tftpboot` και κάτω, ώστε τα πάντα να ανήκουν στο χρήστη μας και στην ομάδα του (pi και pi, για την περίπτωσή μας):

```
pi@nordostbahnhof:~ $ sudo chown -R pi:pi /mnt/external/tftpboot
```

Μεταβαίνουμε στον `/mnt/external/tftpboot`...

```
pi@nordostbahnhof:~ $ cd /mnt/external/tftpboot
```

...και κατεβάζουμε ένα συμπιεσμένο binary της τελευταίας έκδοσης του Memtest86:

```
pi@nordostbahnhof:/mnt/external/tftpboot $ wget \
> http://www.memtest.org/download/5.01/memtest86+-5.01.bin.gz
```

Γιατί το κάναμε αυτό; Μα, γιατί θέλουμε να εξοικειωθούμε με το PXE και τις δυνατότητές του, οπότε καλό είναι να ξεκινήσουμε με ένα απλό παράδειγμα: οι clients θα έχουν επιλογή απομακρυσμένης εκκίνησης του Memtest86 — και για το σκοπό μας πραγματικά δεν έχει σημασία αν υπάρχει ανάγκη ελέγχου της μνήμης τους ή όχι.

Φτιάχνουμε έναν υποκατάλογο για την αποθήκευση του Memtest86 binary, αποσυμπιέζουμε το αρχείο που μόλις κατεβάσαμε και το μετακινούμε εντός του προαναφερθέντος καταλόγου:

```
pi@nordostbahnhof:/mnt/external/tftpboot $ mkdir memtest

pi@nordostbahnhof:/mnt/external/tftpboot $ gzip -d memtest86+-5.01.bin.gz

pi@nordostbahnhof:/mnt/external/tftpboot $ mv memtest86+-5.01.bin \
> memtest/memtest86+-5.01
```

Παρατηρήστε ότι στη νέα θέση του Memtest86 binary, αφήσαμε το ίδιο όνομα αλλά παραλείψαμε την κατάληψη .bin. Τώρα, στο `/mnt/external/tftpboot` δημιουργούμε μια σειρά από symbolic links προς εκείνα τα αρχεία που είναι απαραίτητα για την εκκίνηση του bootloader στους PXE clients:

```
pi@nordostbahnhof:/mnt/external/tftpboot $ ln -s \
> /usr/lib/PXELINUX/pxelinux.0 .

pi@nordostbahnhof:/mnt/external/tftpboot $ ln -s \
> /usr/lib/syslinux/modules/bios/vesamenu.c32 .

pi@nordostbahnhof:/mnt/external/tftpboot $ ln -s \
> /usr/lib/syslinux/modules/bios/ldlinux.c32 .
```

```
pi@nordostbahnhof:/mnt/external/tftpboot $ ln -s \
> /usr/lib/syslinux/modules/bios/libcom32.c32 .

pi@nordostbahnhof:/mnt/external/tftpboot $ ln -s \
> /usr/lib/syslinux/modules/bios/libutil.c32 .
```

Πριν συνεχίσουμε, στον τρέχοντα κατάλογο μεταφέρουμε μια εικόνα για το boot menu. Ο τύπος της πρέπει να είναι PNG και οι διαστάσεις της 640x480 pixels. Δεν χρειάζεται, φυσικά, να δημιουργήσουμε κάτι περίτεχνο. Εμείς, π.χ., φτιάξαμε το αρχείο bootmenu.png, το οποίο είναι μια μαύρη εικόνα με ένα απλό πιγκουινάκι κάτω δεξιά. Ιδού πώς έχουν τα περιεχόμενα του καταλόγου `/mnt/external/tftpboot` έως αυτή τη τη στιγμή:

```
pi@nordostbahnhof:/mnt/external/tftpboot $ ls -lh
total 20K
...  Oct  3 ... bootmenu.png
...  Oct  3 ... ldlinux.c32 -> /usr/lib/syslinux/modules/bios/ldlinux.c32
...  Oct  3 ... libcom32.c32 -> /usr/lib/syslinux/modules/bios/libcom32.c32
...  Oct  3 ... libutil.c32 -> /usr/lib/syslinux/modules/bios/libutil.c32
...  Oct  3 ... memtest
...  Oct  3 ... pxelinux.0 -> /usr/lib/PXELINUX/pxelinux.0
...  Oct  3 ... vesamenu.c32 -> /usr/lib/syslinux/modules/bios/vesamenu.c32
```

Φτιάχνουμε τώρα τον κατάλογο ονόματι `pxelinux.cfg` κι εντός αυτού ένα αρχείο απλού κειμένου, με όνομα `default`. Τα περιεχόμενά του πρέπει να μοιάζουν με τα ακόλουθα:

```
UI vesamenu.c32
PROMPT 0
TIMEOUT 600
DEFAULT Memtest86
MENU TITLE Nordostbahnhof PXE Boot Loader
MENU BACKGROUND bootmenu.png
MENU COLOR SEL 30;47 #000000 #FFFFFFFF

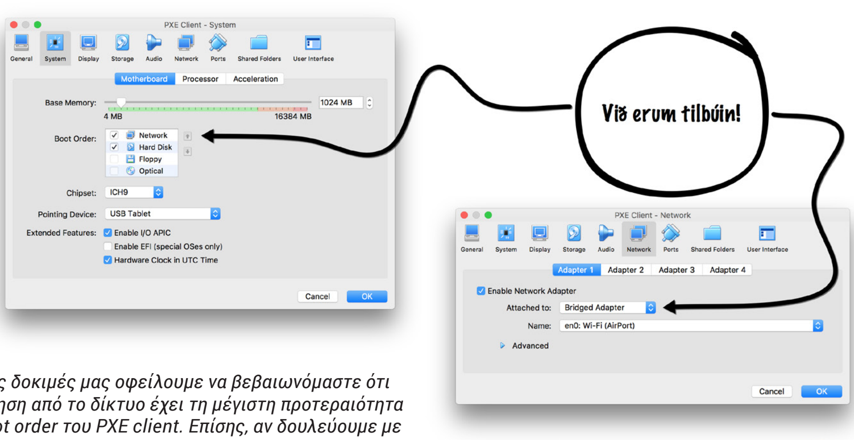
LABEL Memtest86
 MENU LABEL Memtest86+ 5.01
 KERNEL /memtest/memtest86+-5.01
```

Μπορείτε βεβαίως να κάνετε τις τροποποιήσεις σας, αλλά η γενική ιδέα είναι ότι χάρη στο αρχείο `pxelinux.cfg/default` ορίζουμε ένα απλό μενού, το οποίο θα εμφανίζεται στις οθόνες των PXE clients για συγκεκριμένο χρόνο (εκφράζεται σε δέκατα του δευτερολέπτου). Το μενού προς το παρόν έχει μία μόνο επιλογή: το φόρτωμα του Memtest86, για την εκτέλεση διαγνωστικών ελέγχων στη μνήμη RAM του εκάστοτε PXE client.

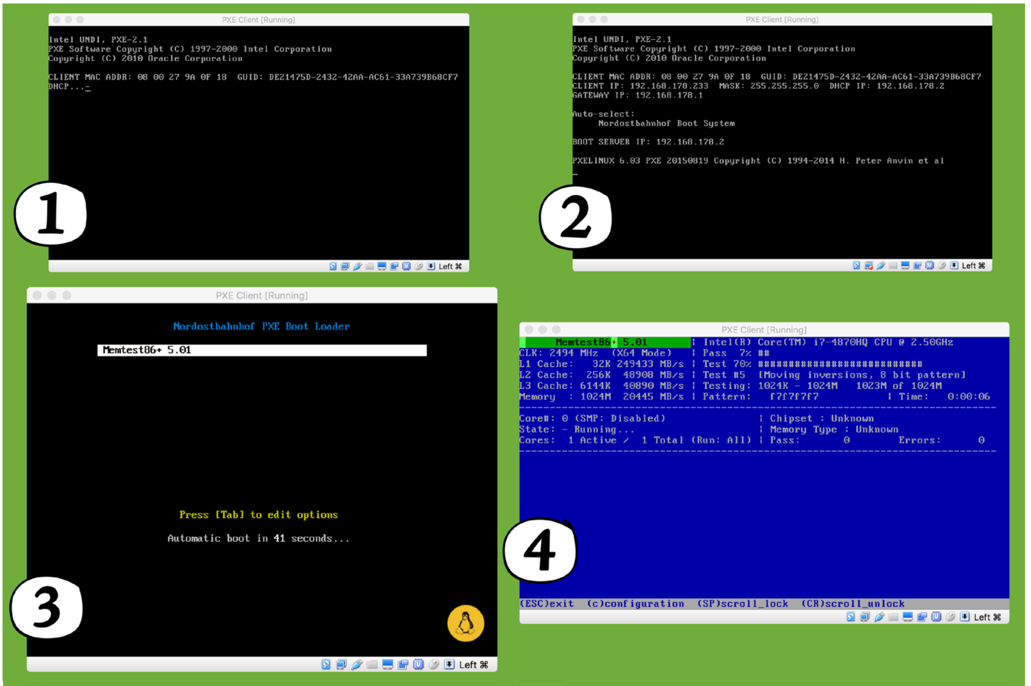
Προκειμένου να βεβαιωθούμε ότι όλα έχουν πάει καλά, δεν έχουμε παρά να ξεκινήσουμε έναν PXE client. Αυτό που θα περιμένουμε να δούμε στην οθόνη του είναι ένα απλό μενού με μία και μόνο επιλογή, για το φόρτωμα του Memtest86. Μόλις το εκτελέσιμο φορτώσει, στον client θα πρέπει να βλέπουμε, ζωντανά, την εξέλιξη μιας σειράς διαγνωστικών ελέγχων για τη μνήμη του. Πρακτικά, δύο είναι οι επιλογές μας.

- Πηγαίνουμε σε έναν φυσικό υπολογιστή του τοπικού με δικτύου και τον ξεκινάμε. Στο boot order, θα πρέπει να φροντίσουμε ώστε πρώτη συσκευή να είναι η κατάλληλη κάρτα δικτύωσης.
- Δημιουργούμε ένα νέο VM με bridged networking, και το ξεκινάμε. Εναλλακτικά στρεφόμαστε σε ένα ήδη υπάρχον VM, βεβαιωνόμαστε ότι έχει bridged networking, και το ξεκινάμε. Όπως και να 'χει, στο boot order του VM πρώτη συσκευή πρέπει να είναι η κάρτα δικτύωσης.

Προσέξτε ότι για τις ανάγκες τις συγκεκριμένης δοκιμής, το αν ο PXE client διαθέτει ή στερείται λειτουργικού συστήματος δεν έχει καμία σημασία. Αυτό που έχει σημασία είναι ότι αφού τον ξεκινήσουμε, μετά από λίγα δευτερόλεπτα θα πρέπει να δούμε στην οθόνη του ένα απλό μενού με μία μόνο επιλογή: το φόρτωμα του Memtest86. Αν δεν δούμε το μενού ή το Memtest86 αποτύχει να φορτώσει, τότε ελέγχουμε το συντακτικό του αρχείου `pxelinux.cfg/default`, τις θέσεις αρχείων και καταλόγων κάτω από τον κατάλογο tftpboot, καθώς και τα symbolic links που φτιάξαμε πριν λίγο. Τέλος, βεβαιωθείτε ότι ο client πράγματι φορτώνει **πρώτα** από την **κατάλληλη** κάρτα δικτύωσης.



Κατά τις δοκιμές μας οφείλουμε να βεβαιωνόμαστε ότι η εκκίνηση από το δίκτυο έχει τη μέγιστη προτεραιότητα στο boot order του PXE client. Επίσης, αν δουλεύουμε με κάποιο VM, η δικτύωση της μηχανής πρέπει να είναι τύπου bridged.



Η πρώτη επιτυχημένη δοκιμή για τον δικό μας PXE server, όπου ελέγχουμε αν κάποιος client είναι ικανός να φορτώσει δικτυακά και να τρέξει το Memtest86.

Περισσότερες επιλογές για PXE booting

Μόλις βεβαιωθήκαμε ότι ο PXE server λειτουργεί. Προς το παρόν προσφέρει στους PXE clients μόνο το Memtest86. Σε λίγο θα επαυξήσουμε τις δυνατότητές του, συνεπώς θα τον καταστήσουμε **πολύ** πιο χρήσιμο για το home network. Στα παραδείγματα που ακολουθούν θα δείξουμε τι ακριβώς χρειάζεται να κάνουμε, ώστε οι PXE clients να έχουν επιλογή για δικτυακή εγκατάσταση δύο δημοφιλών διανομών Linux που πολύ αγαπάμε και χρησιμοποιούμε: του openSUSE Leap 42.3 και του Ubuntu Server 16.04.3 LTS.

Ανεξαρτήτως πάντως των distributions που θα διαθέτει ο PXE server, έχουμε ν' αποφασίσουμε πώς οι PXE clients θα λαμβάνουν τα αρχεία εγκατάστασης. Αναλυτικότερα, ο πυρήνας και το initial ramdisk image κάθε διανομής μεταφέρονται στον εκάστοτε PXE client μέσω TFTP. Τα υπόλοιπα αρχεία, όμως, πρέπει να μεταφέρονται με κάποιον άλλον τρόπο. Σε οδηγούς στο Internet θα δείτε να γίνεται λόγος περί του πρωτοκόλλου NFS. Το δοκιμάσαμε κι εμείς, όμως με συγκεκριμένες διανομές συναντήσαμε προβλήματα κι εμπόδια. Αλλά η χρήση του NFS δεν είναι υποχρεωτική: Κάλλιστα μπορούμε να προσφέρουμε τα αρχεία εγκατάστασης, π.χ., μέσω HTTP. Από τη στιγμή που στο Raspberry Pi μας τρέχει ήδη ο lighttpd –προφέρεται "λάιτι" και τον εγκατέστησε το Pi-hole–, αποφασίσαμε να στρέψουμε την προσοχή μας στη διάθεση μέσω HTTP.

Μεταβαίνουμε στον κατάλογο `/etc/lighttpd/conf-available` και, με δικαιώματα root, δημιουργούμε ένα νέο αρχείο με όνομα `99-distributions.conf` και το ακόλουθο περιεχόμενο:

```
dir-listing.activate = "enable"
```

Μετά πηγαίνουμε στον κατάλογο `/etc/lighttpd/conf-enabled` και δημιουργούμε έναν symbolic link προς το αρχείο που μόλις φτιάξαμε:

```
pi@nordostbahnhof:/etc/lighttpd/conf-enabled $ sudo ln -s \
> ../conf-available/99-distributions.conf .
```

Επανεκκινούμε το lighttpd πληκτρολογώντας `sudo systemctl restart lighttpd`, ώστε να ληφθεί υπόψη η αλλαγή. Μόλις φροντίσαμε ώστε ο web server να παρουσιάζει τα αρχεία και τους υποκαταλόγους των καταλόγων, οι οποίοι **δεν έχουν** αρχεία HTML και βρίσκονται κάτω από το λεγόμενο document root. Στην περίπτωσή μας, αυτό είναι το `/var/www/html`. Ας πάμε λοιπόν σε αυτόν τον κατάλογο (`cd /var/www/html`) κι ας δημιουργήσουμε μια δομή υποκαταλόγων για τη φιλοξενία των αρχείων των διανομών που θέλουμε να παρέχει ο PXE server:

```
pi@nordostbahnhof:/var/www/html $ sudo mkdir -p \
> distributions/osuse-leap-42.3-x86_64

pi@nordostbahnhof:/var/www/html $ sudo mkdir \
> distributions/ubuntu-16.04.3-server-amd64
```

Όλα καλά, μέχρι να συνειδητοποιήσουμε ότι ο κατάλογος `/var/www/html` βρίσκεται στην κάρτα SD του Raspberry. Όποια κι αν είναι η χωρητικότητά της, όσο ελεύθερο χώρο κι αν έχει, σίγουρα δεν θέλουμε ν' αποθηκεύουμε χιλιάδες αρχεία άλλων διανομών σε αυτή. Γι' αυτό και θα εργαστούμε ως εξής:

- θα κατεβάσουμε τα αρχεία ISO των διανομών που θέλουμε
- τα αρχεία ISO θα τα έχουμε σε ένα άλλο μέσο, π.χ., σε έναν εξωτερικό δίσκο USB
- κάθε ISO image θα το προσαρτήσουμε κάτω από τον αντίστοιχο κατάλογο του `/var/www/html/distributions`
- θα προσθέσουμε τα αντίστοιχα entries στο αρχείο /etc/fstab, ώστε οι συνδέσεις (προσαρτήσεις) των ISO να γίνονται αυτόματα κατά την εκκίνηση του Raspbian.

Ας δούμε τα τέσσερα αυτά βήματα αναλυτικότερα.

Κατ' αρχάς, στο πλαίσιο της παρουσίασής μας κατεβάσαμε τα ISO images των openSUSE Leap 42.3 (http://download.opensuse.org/distribution/leap/42.3/iso/openSUSE-Leap-42.3-DVD-x86_64.iso) και Ubuntu Server 16.04.3 LTS (http://ftp.fau.de/ubuntu-releases/16.04.3/ubuntu-16.04.3-server-amd64.iso).

Πήραμε τα αρχεία openSUSE-Leap-42.3-DVD-x86_64.iso και ubuntu-16.04.3-server-amd64.iso, το οποίο και μετονομάσαμε σε Ubuntu-16.04.3-Server-amd64.iso (μη ρωτάτε γιατί, παραξενιές είν' αυτές). Γενικά, όλα τα ISO images τα αποθηκεύουμε στον κατάλογο /mnt/external/ISOs – και τα δύο προαναφερθέντα δεν πήγαν κάπου αλλού.

Για τη χειροκίνητη προσάρτηση των δύο ISO images, ως root πληκτρολογήσαμε:

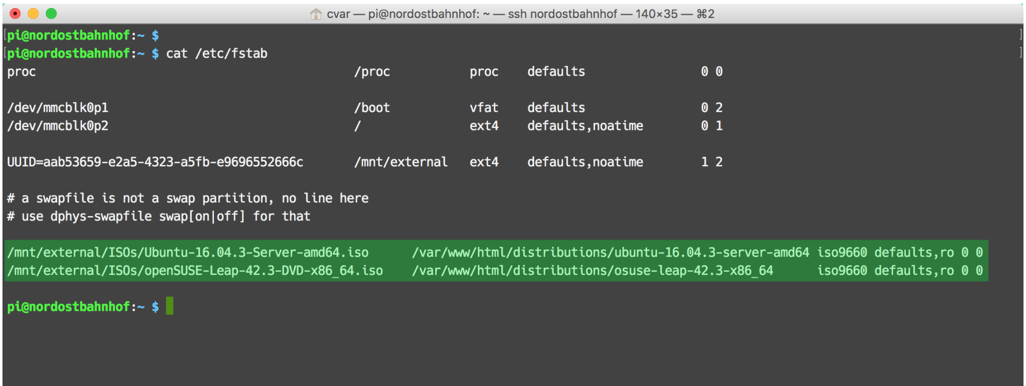
```
pi@nordostbahnhof:~ $ sudo mount \
> /mnt/external/ISOs/openSUSE-Leap-42.3-DVD-x86_64.iso \
> /var/www/html/distributions/osuse-leap-42.3-x86_64
pi@nordostbahnhof:~ $ sudo mount \
> /mnt/external/ISOs/Ubuntu-16.04.3-Server-amd64.iso \
> /var/www/html/distributions/ubuntu-16.04.3-server-amd64
```

```
cvar — pi@nordostbahnhof: ~ — ssh nordostbahnhof — 140×35 — ⌘2
pi@nordostbahnhof:~ $ df -hT
Filesystem     Type      Size  Used Avail Use% Mounted on
/dev/root      ext4       15G  2.4G   12G  17% /
devtmpfs       devtmpfs  481M     0  481M   0% /dev
tmpfs          tmpfs     486M     0  486M   0% /dev/shm
tmpfs          tmpfs     486M   37M  449M   8% /run
tmpfs          tmpfs     5.0M  4.0K  5.0M   1% /run/lock
tmpfs          tmpfs     486M     0  486M   0% /sys/fs/cgroup
/dev/mmcblk0p1 vfat       63M   22M   42M  35% /boot
/dev/sda1      ext4      587G   92G  466G  17% /mnt/external
/dev/loop0     iso9660   825M  825M     0 100% /var/www/html/distributions/ubuntu-16.04.3-server-amd64
/dev/loop1     iso9660   4.4G  4.4G     0 100% /var/www/html/distributions/osuse-leap-42.3-x86_64
pi@nordostbahnhof:~ $
```

Τα ISO images των δύο διανομών που αρχικά σχεδιάζουμε να παρέχουμε στους PXE clients, προσαρτημένα κάτω από τον κατάλογο distributions κι εντός του document root του lighttpd. Η πρόσβαση στα αρχεία εγκατάστασης των διανομών θα γίνεται μέσω HTTP.

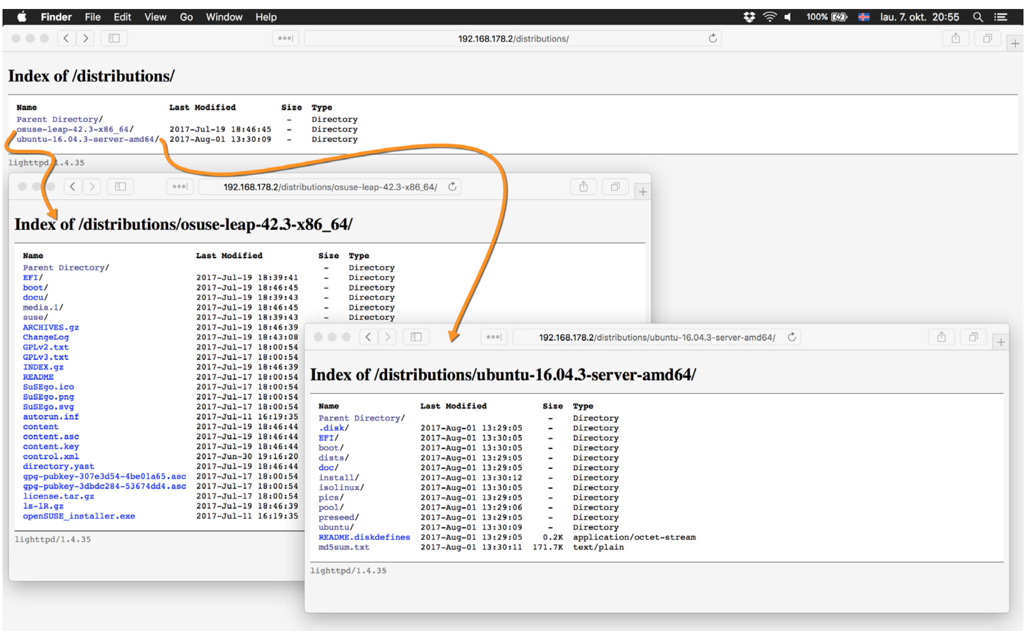
Θέλουμε τα ISO images να προσαρτώνται αυτόματα κατά την εκκίνηση του Raspbian, γι' αυτό και προσθέτουμε τις κατάλληλες γραμμές στο /etc/fstab. Δείτε στο ακόλουθο screenshot τις σχετικές γραμμές που έχουμε βάλει στο δικό μας fstab.

Προκειμένου να βεβαιωθούμε ότι παρέχεται πρόσβαση HTTP στο περιεχόμενο των δύο ISO images, από έναν web browser, σε ένα οποιοδήποτε μηχάνημα του τοπικού δικτύου, αρκεί να επισκεφτούμε τις διευθύνσεις http://192.168.178.2/



Σε πρασινωπό υπόβαθρο είναι οι δύο γραμμές στο /etc/fstab του Raspberry Pi μας, οι οποίες διασφαλίζουν την αυτόματη προσάρτηση των ISO images των διανομών που προσφέρουμε στους PXE clients του τοπικού δικτύου.

distributions, http://192.168.178.2/distributions/osuse-leap-42.3-x86_64 και βεβαίως http://192.168.178.2/distributions/ubuntu-16.04.3-server-amd64 (στη θέση του 192.168.178.2 θα βάλετε, φυσικά, τη διεύθυνση IP του RasPi σας).



Έλεγχος για την πρόσβαση στο περιεχόμενο των προσαρτημένων ISO images μέσω HTTP, και χάρη στον lighttpd web server που έχει εγκαταστήσει το Pi-hole.

Ωραία. Είδαμε έναν τρόπο για παροχή πρόσβασης στα αρχεία εγκατάστασης των διανομών, μέσω HTTP. Στη συνέχεια φροντίζουμε ώστε οι PXE clients να έχουν δυνατότητα για επιλογή διανομής και βεβαίως για δικτυακή εγκατάστασή της.

Περίπτωση openSUSE

Πηγαίνουμε κάτω από το `/mnt/external/tftpboot` και φτιάχνουμε μια δομή καταλόγων για την αποθήκευση του πυρήνα και του initial ramdisk image **κάθε** διανομής που σχεδιάζουμε να προσφέρουμε μέσω PXE. Για το openSUSE Leap 42.3, εμείς δώσαμε:

```
pi@nordostbahnhof:/mnt/external/tftpboot $ mkdir -p \
> boot/osuse-leap-42.3-x86_64
```

Μέσα στον `boot/osuse-leap-42.3-x86_64` αντιγράφουμε τα αρχεία `linux` και `initrd`, από τον κατάλογο `/var/www/html/distributions/osuse-leap-42.3-x86_64/boot/x86_64/loader`. Με έναν text editor ανοίγουμε το αρχείο `pxelinux.cfg/default` και του προσθέτουμε τις ακόλουθες γραμμές (αλλάξτε IP και paths, ώστε να ανταποκρίνονται στο setup σας):

```
LABEL openSUSE-Leap-42.3
 MENU LABEL openSUSE Leap 42.3
 KERNEL boot/osuse-leap-42.3-x86_64/linux
 APPEND initrd=boot/osuse-leap-42.3-x86_64/initrd install=http://192.168.178.2/
distributions/osuse-leap-42.3-x86_64
```

(**Ολόκληρη** η τελευταία γραμμή είναι: `APPEND initrd=boot/osuse-leap-42.3-x86_64/initrd install=http://192.168.178.2/distributions/osuse-leap-42.3-x86_64`.) Αποθηκεύουμε τις αλλαγές, εγκαταλείπουμε τον editor και είμαστε έτοιμοι να δοκιμάσουμε την πρώτη μας δικτυακή εγκατάσταση μέσω PXE boot. Δεν είναι απαραίτητο να επανεκκινήσουμε τον dnsmasq, τον lighttpd ή κάποια άλλη υπηρεσία στο RasPi. Απλά, χρειαζόμαστε έναν φυσικό ή εικονικό υπολογιστή που ξεκινά ψάχνοντας για εκτελέσιμο κώδικα **πρώτα** μέσω της PXE capable κάρτας δικτύου που διαθέτει. Αν όλα έχουν πάει καλά, στην οθόνη του client θα δούμε το απλό μενού που ορίζεται από το `pxelinux.cfg/default` και προς το παρόν περιλαμβάνει δύο entries: `Memtest86+ 5.01` και `openSUSE Leap 42.3`. Επιλέγοντας το δεύτερο, στο κάτω μέρος της οθόνης θα παρατηρήσουμε ότι πρώτα φορτώνει ο πυρήνας, μετά το initial ramdisk image. Ίσως αυτό το πρώτο στάδιο της εγκατάστασης σας φανεί ότι διαρκεί περισσότερο από το αναμενόμενο. Μην ανησυχείτε, αυτό είναι εντελώς φυσιολογικό. Μετά από λίγο θα ξεκινήσει ο installer του openSUSE και θα μπορέσετε να φέρετε εις πέρας την εγκατάσταση του λειτουργικού. Θα σας προτείναμε να το κάνετε, ώστε να είστε 100% βέβαιοι ότι όλα είναι όπως πρέπει.

Περίπτωση Ubuntu

Κατά τα αναμενόμενα, εργαζόμαστε παρόμοια με την περίπτωση του openSUSE. Κάτω από το `/mnt/external/tftpboot/boot` φτιάχνουμε τον κατάλογο `ubuntu-16.04.3-server-amd64`. Μέσα στον νέο αυτό κατάλογο αντιγράφουμε τα αρχεία linux και initrd.gz, από τη θέση `/var/www/html/distributions/ubuntu-16.04.3-server-amd64/install/netboot/ubuntu-installer/amd64`.

Μετά προσθέτουμε τις ακόλουθες γραμμές στο αρχείο `pxelinux.cfg/default`:

```
LABEL Ubuntu-16.04.3-Server-LTS
 MENU LABEL Ubuntu 16.04.3 Server LTS
 KERNEL boot/ubuntu-16.04.3-server-amd64/linux
 APPEND initrd=boot/ubuntu-16.04.3-server-amd64/initrd.gz
install=http://192.168.178.2/distributions/ubuntu-16.04.3-server-amd64
```

(**Ολόκληρη** η τελευταία γραμμή είναι: `APPEND initrd=boot/ubuntu-16.04.3-server-amd64/initrd.gz install=http://192.168.178.2/distributions/ubuntu-16.04.3-server-amd64`.) Ξεκινήστε έναν φυσικό ή εικονικό PXE client και στο μενού με τα συστήματα προς εγκατάσταση θα δείτε τρεις γραμμές. Μετακινηθείτε πάνω από την Ubuntu 16.04.3 Server LTS και πιέστε το [Enter]. Αμέσως θα ξεκινήσει το φόρτωμα του πυρήνα και μετά του `initial ramdisk image`. Αν έχετε ήδη δοκιμάσει την εγκατάσταση του `openSUSE Leap 42.3` μέσω PXE, όπως δείξαμε πριν λίγο, τώρα θα διαπιστώσετε ότι ο installer του Ubuntu θα ξεκινήσει αισθητά πιο γρήγορα. Δοκιμάστε να προχωρήσετε με την εγκατάσταση. Αν όλα έχουν πάει καλά, θα ολοκληρωθεί ομαλά και χωρίς προβλήματα.

Η συνέχεια

Από τη στιγμή που έχουμε κατανοήσει τη λογική του PXE booting μέσω TFTP, το μόνο που πρέπει να προσέχουμε είναι τη δομή καταλόγων εντός του `tftpboot`. Είναι σημαντικό να έχουμε μια καθαρή εικόνα για τις θέσεις των καταλόγων που φιλοξενούν kernel και initial ramdisk images, των διανομών που θέλουμε να προσφέρει ο PXE server.

Προσοχή χρειάζεται επίσης η σύνταξη νέων επιλογών για το μενού που παρουσιάζει ο server στους PXE clients. Αναφερόμαστε στο αρχείο `pxelinux.cfg/default`: αν υπάρχει κάποιο λάθος πληκτρολόγησης για κάποια επιλογή, το φόρτωμα πυρήνα ή/και initial ramdisk image απλά θα αποτυγχάνει **χωρίς** κατ' ανάγκη να παίρνουμε κάποιο σχετικό μήνυμα στην οθόνη.

Τέλος, νομίζουμε ότι έχει ήδη γίνει φανερό ότι χρειάζεται μια κάποια προσοχή για την επιλογή των αρχείων πυρήνα και ramdisk image, για κάθε διανομή που θέλουμε να προσφέρει ο PXE server. Ως άσκηση, προτείνουμε να φροντίσετε ώστε ο δικό σας PXE server να παρέχει δυνατότητα εγκατάστασης των πλέον πρόσφατων εκδοχών των Debian και Fedora.

Καλή επιτυχία σας ευχόμαστε – και καλή σας διασκέδαση!

```
cvar — pi@nordostbahnhof: /mnt/external/tftpboot — ssh nordostbahnhof — 140×34 — ⌘2
pi@nordostbahnhof:/mnt/external/tftpboot $
pi@nordostbahnhof:/mnt/external/tftpboot $ tree
.
├── boot
│   ├── osuse-leap-42.3-x86_64
│   │   ├── initrd
│   │   └── linux
│   └── ubuntu-16.04.3-server-amd64
│       ├── initrd.gz
│       └── linux
├── bootmenu.png
├── ldlinux.c32 -> /usr/lib/syslinux/modules/bios/ldlinux.c32
├── libcom32.c32 -> /usr/lib/syslinux/modules/bios/libcom32.c32
├── libutil.c32 -> /usr/lib/syslinux/modules/bios/libutil.c32
├── memtest
│   └── memtest86+-5.01
├── pxelinux.0 -> /usr/lib/PXELINUX/pxelinux.0
├── pxelinux.cfg
│   └── default
└── vesamenu.c32 -> /usr/lib/syslinux/modules/bios/vesamenu.c32

5 directories, 12 files
pi@nordostbahnhof:/mnt/external/tftpboot $
```

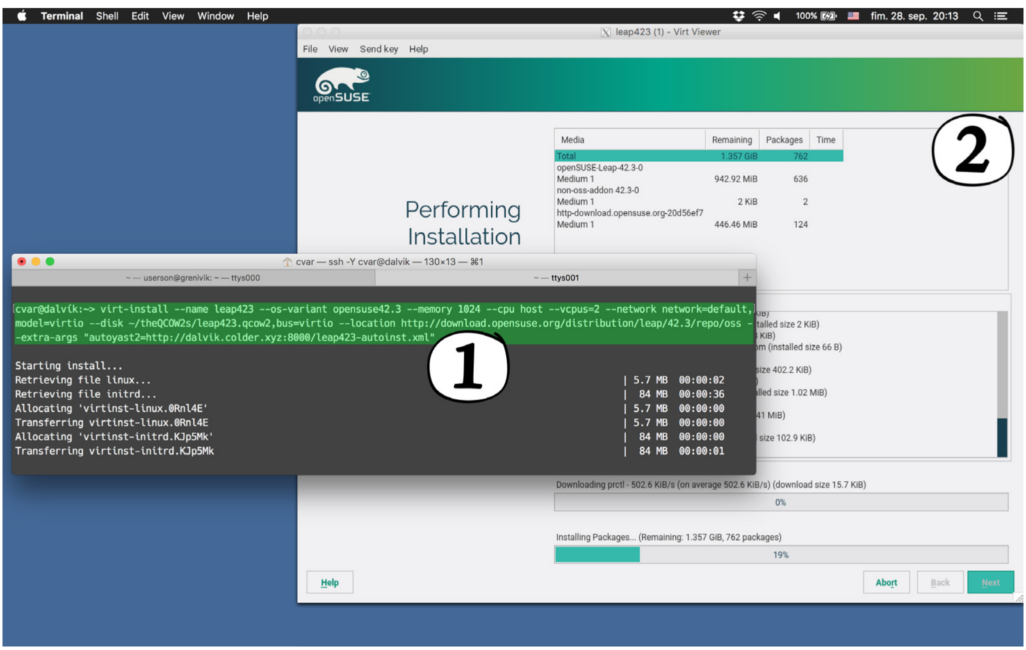
Η δομή καταλόγων και τα αρχεία κάτω από το `tftpboot`, στο πλαίσιο της παρουσίασής μας. Προς το παρόν ο PXE server μας προσφέρει δυνατότητα εγκατάστασης δύο διανομών στους clients, όμως εύκολα μπορούμε να εμπλουτίσουμε το μενού του.


CMYK & WEB
cmykandweb.gr
GRAPHIC DESIGN
design • layout • web

Skill: Intermediate
Tags: AutoYaST, KVM, libvirt, openSUSE, QEMU

Άνετα κι αστραπιαία: Προετοιμασία openSUSE VM

Έχουμε ήδη δείξει πώς δημιουργούμε εικονικές μηχανές με το openSUSE Leap 42.3, ταχύτατα κι από τη γραμμή εντολών. Όσοι έχετε διαβάσει το σχετικό άρθρο, τώρα θα εκπλαγείτε: Ναι, η προετοιμασία του VM είναι δυνατόν να γίνει ακόμη πιο γρήγορα!



Από έναν υπολογιστή με OS X έχουμε συνδεθεί μέσω SSH σε έναν headless Linux server, ο οποίος εκτελεί χρέη virtualization host. Πληκτρολογούμε ό,τι είναι να πληκτρολογήσουμε στο τερματικό, οπότε αμέσως δημιουργείται ένα νέο VM και ξεκινά η αυτοματοποιημένη εγκατάσταση του openSUSE Leap 42.3. Την πρόοδο της όλης διαδικασίας παρακολουθούμε από το παράθυρο που ανοίγει ο Virt Viewer.

Αναφερόμαστε σε QEMU/KVM VMs και φυσικά δουλεύουμε σε περιβάλλον Linux. Με μία εντολή που μοιάζει με την παρακάτω, δημιουργούμε τη μηχανή και ξεκινάμε την εγκατάσταση του openSUSE Leap 42.3. Η όλη διαδικασία είναι εντελώς αυτοματοποιημένη κι από μέρους μας δεν χρειάζεται να κάνουμε κάτι.

```
cvar@dalvík:~> virt-install --name leap423 --os-variant opensuse42.3 \
--memory 1024 --cpu host --vcpus=2 \
--network network=default,model=virtio \
--disk ~/theQCOW2s/leap423.qcow2,bus=virtio \
--location http://download.opensuse.org/distribution/leap/42.3/repo/oss \
--extra-args "autoyast2=http://dalvik.colder.xyz:8000/leap423-autoinst.xml"
```

Πριν αρχίσετε τη δημιουργία της μηχανής και την εγκατάσταση του guest OS, κάντε παρακαλούμε λίγη υπομονή και δώστε προσοχή στα τέσσερα σημεία που ακολουθούν.

- Στο παράδειγμά μας, η δικτύωση του VM βασίζεται στην τεχνική του NAT Forwarding. Αν θέλετε Bridged Networking, τότε στη θέση του `--network network=default,model=virtio` βάλτε το `--network bridge=br0,model=virtio`.
- Ο εικονικός δίσκος της μηχανής είναι το αρχείο `leap432.qcow2` και βρίσκεται εντός του pool ονόματι `theQCOW2s`, κάτω από τον προσωπικό κατάλογο του χρήστη μας. Διαβάστε στο https://deltahacker.gr/?p=17646 πώς δημιουργούμε pools και δίσκους για VMs.
- Η εγκατάσταση του openSUSE Leap 42.3 γίνεται δικτυακά, χωρίς να κατεβάσουμε κάποιο ISO image. Γι' αυτό και στην παράμετρο `--location` δώσαμε ένα URL προς δικτυακή τοποθεσία όπου φιλοξενούνται όλα τα απαραίτητα αρχεία για μια **πλήρη** εγκατάσταση της διανομής.
- Η ίδια η διαδικασία της εγκατάστασης γίνεται **εντελώς** αυτοματοποιημένα, χάρη στο AutoYaST. Στο GRUB2 περνάμε τη σχετική οδηγία με τη βοήθεια της παραμέτρου `--extra-args`. Αν δεν έχετε κάποιο AutoYaST profile κι ανυπομονείτε να ξεκινήσετε τους πειραματισμούς, πάρτε το αρχείο `leap423-autoinst.xml` απ' αυτό το σεμνό repository: https://github.com/colder-is-better/autoyast. Περί AutoYaST διαβάστε στο https://deltahacker.gr/?p=17580 **όλα** όσα πρέπει να γνωρίζετε.

Ξεχάσαμε κάτι; Μάλλον όχι, αλλά αν κάνουμε λάθος βάλτε φωνή.

deltahacker.gr